

Acquiring user tradeoff strategies and preferences for negotiating agents: A default-then-adjust method

Xudong Luo*, Nicholas R. Jennings, Nigel Shadbolt

Intelligence, Agents, Multimedia Group, School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

Received 3 March 2004; received in revised form 1 June 2005; accepted 4 August 2005

Available online 19 September 2005

Communicated by E. Motta

Abstract

A wide range of algorithms have been developed for various types of negotiating agents. In developing such algorithms the main focus has been on their efficiency and their effectiveness. However, this is only a part of the picture. Typically, agents negotiate on behalf of their owners and for this to be effective the agents must be able to adequately represent their owners' strategies and preferences for negotiation. However, the process by which such knowledge is acquired is typically left unspecified. To address this problem, we undertook a study of how user information about negotiation tradeoff strategies and preferences can be captured. Specifically, we devised a novel *default-then-adjust* acquisition technique. In this, the system firstly does a structured interview with the user to suggest the attributes that the tradeoff could be made between, then it asks the user to adjust the suggested default tradeoff strategy by improving some attribute to see how much worse the attribute being traded off can be made while still being acceptable, and, finally, it asks the user to adjust the default preference on the tradeoff alternatives. This method is consistent with the principles of standard negotiation theory and to demonstrate its effectiveness we implemented a prototype system and performed an empirical evaluation in an accommodation renting scenario. The result of this evaluation indicates the proposed technique is helpful and efficient in accurately acquiring the users' tradeoff strategies and preferences.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Tradeoff strategy and preference; Knowledge acquisition; Preference acquisition; Automated negotiation; Software agents

1. Introduction

Negotiation—the process by which a group of agents come to a mutually acceptable agreement on some matter—is a key form of interaction in a wide range of complex systems (including the Web, the Grid, peer-to-peer systems, pervasive computing, and e-business). Given its importance, a wide range of negotiation models have been developed; these include models for auctions, direct one-to-one negotiations, and argumentation-based encounters (see He et al., 2003; Lomuscio et al., 2003 for an overview). To date, however, research in this field has been almost exclusively concerned with the development of efficient and effective algorithms that enable agents to be

successful and obtain acceptable outcomes. While this is clearly important, it is only part of the picture. In most cases, agents negotiate on behalf of their owner (which may be an individual or an organisation). Now for this to be effective, agents must be able to adequately reflect their owners' interests, strategies, preferences, and prejudices in the given domain. Without this, software agents cannot be delegated negotiation tasks. So, the acquisition of such knowledge is an essential prerequisite for applying negotiating agents in practice. However, at this time, little thought has been given to the problems of determining: (i) exactly what knowledge an owner needs to impart to their agent in order to achieve high fidelity negotiation behaviour, and (ii) how such knowledge can be effectively acquired from the owner. These are clearly serious shortcomings of existing research that need to be addressed if negotiating agents are to be widely used.

*Corresponding author.

E-mail addresses: xl@ecs.soton.ac.uk (X. Luo), nrj@ecs.soton.ac.uk (N.R. Jennings), nrs@ecs.soton.ac.uk (N. Shadbolt).

Against this background, our research seeks to start bridging the knowledge acquisition gap between the negotiating agents' owners and the negotiation algorithms that their agents use. Specifically, in this paper we focus on exploring how the necessary knowledge about a user's negotiation tradeoffs can be acquired. In more detail, the term *tradeoff* among negotiation attributes refers to combinations of attributes' values in which the worsening of some attributes is used to get other attributes' values improved. The set of all such potential combinations is called the user's *tradeoff strategy* and its individual elements are called *tradeoff alternatives*. Some of these alternatives are more preferred by the users and this ordering is here referred to as a user's *tradeoff preference*. In this work, such a preference is realised through assigning each tradeoff alternative a satisfaction degree, where the more preferred an alternative is, the higher the corresponding satisfaction degree. Moreover, we focus, in particular, on tradeoffs between pairs of attributes. For example, in an accommodation renting scenario (which we will use as our running example throughout this paper), the following could be a prospective tenant's tradeoff strategy between rental rate and period: (i) "The limit of the rental period I am willing to accept is 1 week if the rental rate is £325 per calendar month (pcm)", (ii) "I am willing to increase my rental period limit to 6 months if I can have the rental rate of £265 pcm", and (iii) "I am willing to increase my rental period limit to 12 months if I can have the rental rate of £250 pcm". Moreover, the user may have a preference on these tradeoff alternatives: thus alternative (i) might be the least preferred, and alternative (iii) the most preferred.

We choose to focus on tradeoffs because they are a fundamental aspect of negotiation in both human behaviour (e.g., the principled negotiation approach (Fisher and Ury, 1981)) and in the behaviour of software agents (e.g., Mudgal and Vassileva, 2000; Faratin et al., 2002; Cheung et al., 2003; Luo et al., 2003b; Zhu, 2004). Moreover, since negotiation participants are normally assumed to be maximisers of their interest, it is highly unlikely that each side in a negotiation will be able to get a deal with the best values of all attributes and, therefore, tradeoffs are inevitable (Raiffa, 1982). For example, it is normally impossible for buyers to get a product that is both cheap and of high quality, or delivered immediately but highly tailored to the individual's particular preferences. In this case, the negotiation participants attempt to minimise the loss of their overall interest (Fisher and Ury, 1981; Raiffa, 1982). Now this can be achieved by making tradeoffs between the different attributes. Such tradeoffs become feasible when the participants have different opinions about which attributes are more important. For example, for the attributes *price* and *quantity*, it is possible that *price* is more important for a buyer but less so for a seller; while *quantity* is less important for the buyer but more so for the seller. Then, the buyer might be willing to concede on *quantity* (i.e., buy more) in order to get an agreement with the seller (i.e., buy more if cheaper), and the seller might be

willing to concede on *price* (i.e., discount more) to get the buyer to concede on *quantity* (i.e., cheaper if buy more). Through such behaviour, win-win solutions can be reached (Fisher and Ury, 1981).

In order to find possible tradeoff alternatives, one could naïvely construct a value function that is consistent with an ordering on a small number of tradeoff alternatives (e.g., some value combinations of attributes *distance* and *rental rate* in our accommodation renting scenario). Then, any other tradeoff alternatives could be computed simply by looking for attribute value combinations that have the same preference level (e.g., in the accommodation renting scenario, different distances have different utility values and so do different rental rates, but their overall utility value obtained using the chosen value function may have the same value (Perloff, 1998)). This could be achieved by arbitrarily picking some attributes and then start worsening or improving their values while maintaining the overall preference level. However, for this to work, the following steps are needed: (i) acquire the user's preferences on values of individual attributes (e.g., the utility function of attribute *distance* in the accommodation renting scenario); (ii) ask users to assign a satisfaction degree to a number of value combinations of individual attributes (e.g., some value pairs of attributes *distance* and *rental rate*); and (iii) find a value function that can aggregate the preferences on the individual attribute values to produce overall preference levels that are consistent with users' satisfaction degrees for the attribute value combinations. Unfortunately, however, in many situations, the user may be unwilling or unable to express their preference on single attribute's values. For example, when renting some accommodation, it is difficult to say which rental rate is preferred more because it depends on other attributes of the property such as distance, size, Internet connection, and so on. Rather, it is much easier to indicate which combination of, say, rental rate and distance, they prefer more (which is the way the method we propose in this paper works). There are also a number of other problems associated with the above approach: (i) there are many aggregation functions that could be used to obtain the overall preference levels from the individual attributes' preference levels (Dubois and Prade, 2004) and it is not obvious how a user could choose between them (Beliakov and Warren, 2001); and (ii) it is not obvious how a value function can be updated in the face of new inconsistent data.

To overcome these problems with the naïve approach, our acquisition method works in the following way. First, conduct a structured interview with the user in order to suggest the attributes that the tradeoff could or should be made between. Second, ask users to adjust the suggested default curve of the tradeoff strategy by improving some attribute to see how much worse the attribute being traded off can be made while still being acceptable. For example, what is the maximum distance away from study that the renter can accept if the estate agency could decrease the

rental rate from £290 to £230 pcm. Finally, adjust the suggested default preference on tradeoff alternatives by adjusting the parameters that configure the preference.

This *default-then-adjust* approach is chosen for a number of reasons. First, in many negotiation scenarios it is not obvious between which negotiation attributes a tradeoff should be made. For example, when people buy their first second-hand car, they often have little idea between which attributes (among model, price, colour, mileage, brand, manual or automatic, and so on) the tradeoffs could or should be made. In this situation, they need the help of an adviser who has the appropriate knowledge. The system developed in this paper functions as just such an adviser: after asking a number questions, it suggests the attributes between which tradeoffs can be made. Second, according to standard business negotiation theory (Fisher and Ury, 1981; Pruitt, 1981; Raiffa, 1982; Unt, 1999), its users should be allowed to explore the space of tradeoff alternatives as exhaustively as they are willing to do. Now, one common technique for doing this is to play simulated negotiation games in which various negotiation scenarios are enacted against various human negotiation partners. So, we design the acquisition system to play the role of a negotiating opponent which allows the user to adjust the default tradeoff strategies and preferences as often or as much as they wish. Third, we want to minimise the user's workload, but maximise the flexibility to express complex tradeoff strategies and preferences where they exist. In our preliminary work in this area we develop an exhaustive method (Luo et al., 2004) (much like the naïve one), but it had the disadvantage that it places a significant burden of work on its users. To counteract this, we developed a knowledge based method (Luo et al., 2003a) which lessens the user's workload, but has the disadvantage that its users cannot flexibly specify their tradeoff strategies (actually the system chooses tradeoff strategies for the user from a *fixed* number of *pre-determined* strategies according to its analysis of the user's situation). The default-then-adjust method described in this paper is an amalgam of these two, retaining both their benefits and removing their disadvantages. Thus, its users get low workload (by setting default tradeoff strategies and preferences) and flexibility to express their individuality (by allowing them to adjust these defaults).

In developing our method, this paper advances the state of the art in a number of ways. First, although tradeoffs are an essential aspect in negotiation, the problem of acquiring information about them has received little attention in the research areas of general knowledge acquisition and preference elicitation. Given the importance of the problem, our work can be viewed as an initial contribution to these areas. Second, the limitations of the existing work in this area (detailed more in Section 6) have been removed. Actually, the biggest advantage of this approach is the use of *default* knowledge. If this default knowledge is close to either the user's tradeoff strategy or preference then the number of steps needed for adjustment is small. Indeed, the

use of such domain knowledge has been found to be effective in related study (e.g., the work of Haddawy et al. (2003) uses domain knowledge in a similar manner for value function elicitation, although the adjustment process is not via a graphical user interface like ours). Third, we have developed and evaluated a knowledge acquisition tool for this task for the domain of accommodation renting. Our trials show that users find it easy to understand and operate, find it helpful to specify their tradeoff strategies and preferences, and find it efficient in terms of the amount of work involved.

The remainder of this paper is structured as follows. Section 2 presents the formal definition of tradeoff strategies and preferences and provides a number of concrete examples of their use. Section 3 outlines the acquisition method to obtain users' tradeoff strategies and preferences. Section 4 presents the associated tool we have developed and shows how it was used in an accommodation renting scenario. Section 5 presents an empirical evaluation of our method and the tool. Section 6 discusses related work. Finally, Section 7 concludes and outlines avenues of further research.

2. Representing user tradeoff strategies and preferences

In this section, we define notions of users' strategies and preferences with respect to negotiation tradeoffs between pairs of attributes that are the most common (Steel and Beasor, 1999; Collin, 2001).¹ Specifically, they can be formally defined as:

Definition 1. Let attribute x take values on $X = [l_x, u_x]$ (where l_x is the best value of x and u_x is the worst one), and attribute y take values on $Y = [l_y, u_y]$ (where l_y is the best value of y and u_y the worst). Then a mapping $t : X \rightarrow Y$ is a tradeoff strategy between x and y if:

- (1) continuity: $\forall x \in X, \forall \varepsilon > 0, \exists \delta > 0, |x - x'| < \delta \Rightarrow |t(x) - t(x')| < \varepsilon$;
- (2) monotonicity: $\forall x, x' \in X, x \geq x' \Leftrightarrow t(x) \leq t(x')$; and
- (3) boundary condition: $t(l_x) = u_y, t(u_x) = l_y$.

Thus, $\forall x \in X, (x, t(x))$ is a tradeoff alternative which means that if attribute y takes the value of $t(x)$, the most the user can accept is x .

In the above definition, the continuity assumption is based on the intuition that when a variable changes its value gradually, the value being traded off should change in a similarly gradual manner (Mas-Collell et al., 1995; Luo and Zhang, 1999); the monotonicity condition reflects the essential meaning of a tradeoff: if one attribute is getting

¹Experience indicates that most tradeoffs are between pair of attributes (Steel and Beasor, 1999; Collin, 2001) and so this is the focus of this work. Having said this, it would certainly be interesting to investigate the case of making tradeoffs between more than two attributes, but we leave this as future work.

better than another one should get worse (Peasall, 2001; Collin, 2001); and the boundary condition means that to get the best value of one attribute, the other has to be made the worst (this assumption is made because the curve of a tradeoff strategy needs to be closed at its left and right ends as shown in Figs. 1–3).

Intuitively, for the same value $t(x)$ of attribute y , different users might have different opinions about how much worse the value of attribute x (being traded off) can be made while still being acceptable. By means of illustration, consider the following examples.

First, if attribute x is much more important to a user than attribute y , then when he uses y 's good values to tradeoff x 's very bad ones, he might be willing to make y a lot worse in order to get x a little better and, moreover, the

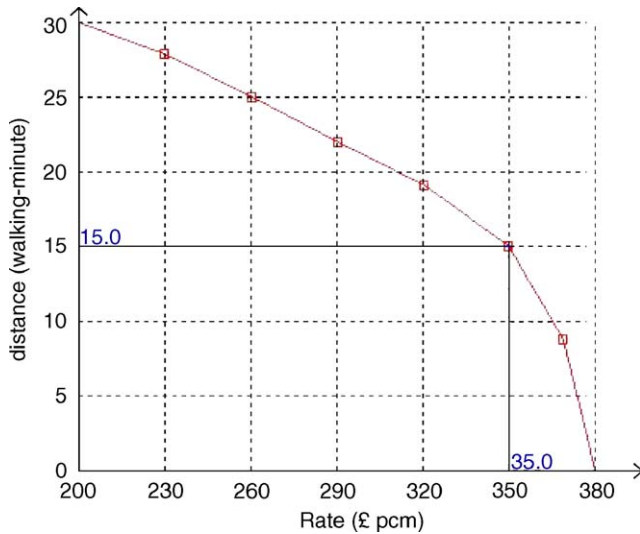


Fig. 1. Increasing distance considerably to get very high rental rate down a little.

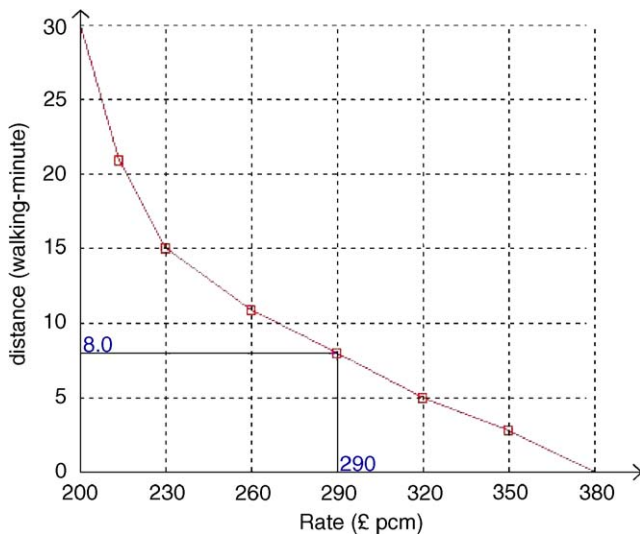


Fig. 2. Increasing distance a little to get very high rental rate down a lot.

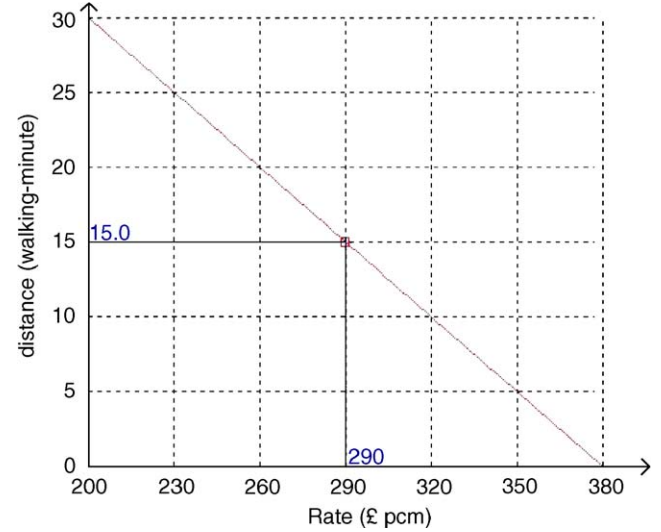


Fig. 3. Increasing distance to the same extent to which rental rate goes down.

worse x is the more he needs to worsen y . For example, in our accommodation renting scenario (see Fig. 1), if the student values the rental rate much more than the walking distance, then when the rental rate is £380 pcm (high) and the walking distance is insignificant, he might be willing to increase the walking distance to 15 min (half way to the worst value, or say $\frac{30-15}{30} = 50\%$ worse) to get the rental rate reduced to £350 pcm ($\frac{380-350}{380-200} = 16.7\%$ better).

Second, if attribute x is much less important to a user than attribute y , then when he uses y 's good values to tradeoff x 's very bad ones, he might be willing to make y just a little worse in order to get x a lot better and, moreover, the better y 's value is, the less he needs to worsen y in order to get x better. For example, if the rental rate is less important to the student than the walking distance (Fig. 2), then when the rental rate is £380 pcm and the walking distance is insignificant, he might be willing to increase the walking distance to 8 min ($\frac{8}{30} = 26.7\%$ worse) in order to reduce the rate to £290 pcm ($\frac{380-290}{380-200} = 50\%$ better).

Third, if attribute x is equally as important as attribute y , then the user may be willing to worsen y to the same extent to which x is made better. Thus, if the rental rate is as important as the walking distance (Fig. 3), then when the rental rate is £380 pcm and the walking distance is insignificant, the user might be willing to increase the walking distance to 15 min ($\frac{15}{30} = 50\%$ worse) in order to reduce the rental rate to £290 pcm ($\frac{380-290}{380-200} = 50\%$ better).

Now, we turn to the concept of tradeoff preference. First, its formal definition is given.

Definition 2. Suppose t is a tradeoff strategy for worsening attribute $y \in [l_y, u_y]$ in order to improve attribute $x \in [l_x, u_x]$. Let $l_x \leq a \leq b \leq c \leq d \leq u_x$ and $T = \{(x, t(x) \mid$

$x \in [l_x, u_x]$. A mapping $p: T \rightarrow [0, 1]$ is a tradeoff preference on T if $\forall x, x' \in [l_x, u_x]$:

- (1) complete dissatisfaction: $\forall x \in [l_x, a] \cup [d, u_x], p(x, t(x)) = 0$;
- (2) satisfaction increasing: $\forall x, x' \in [a, b], x \leq x' \Rightarrow p(x, t(x)) \leq p(x', t(x'))$;
- (3) complete satisfaction: $\forall x \in [b, c], p(x, t(x)) = 1$; and
- (4) satisfaction decreasing: $\forall x, x' \in [c, d], x \leq x' \Rightarrow p(x, t(x)) \geq p(x', t(x'))$.

Here the preference value of a tradeoff alternative is the user's degree of satisfaction with the alternative. In particular, $p(x, t(x)) = 0$ reflects the fact that the user is completely dissatisfied with $(x, t(x))$; $0 < p(x, t(x)) < 1$ reflects the fact that the bigger the value of $p(x, t(x))$ the more the user is satisfied with $(x, t(x))$; and $p(x, t(x)) = 1$ reflects the fact that the user is completely satisfied with $(x, t(x))$.

The intuition behind these properties listed in the above definition are as follows. *First*, although worsening attribute y can lead to an improvement in attribute x , a user may have a limit $t(a)$ for the worst value of attribute y beyond which he cannot go. That is, if attribute y is worse than $t(a)$, although attribute x could be made better than a , the buyer is completely dissatisfied with $(x, t(x))$ when $x \leq a$. Similarly, although attribute x can be improved by worsening attribute y , a buyer may have a limit $t(a)$ for the worst value of attribute x , beyond which he cannot go. *Second*, the case in which attribute y is beyond its maximum limit is completely unacceptable. Thus, when attribute y is getting better (although attribute x is getting worse a little) he should be increasingly satisfied. *Third*, when the user's satisfaction increases to a particular point $(b, t(b))$, both attributes x and y are acceptable and he is completely satisfied. *Fourth*, the case in which attribute x is beyond its maximum limit (e.g., the budget of price) is completely unacceptable. Thus when attribute x is getting better by worsening attribute y , he should be increasingly satisfied. In short, the preference ordering induced by satisfaction degrees is a (potentially) piecewise nonmonotonic function consisting of 5 segments.²

To illustrate these concepts, consider the following example from our accommodation renting scenario. Suppose $d = t(r)$ is a tradeoff strategy between *rate* r and *distance* d , then the following is a preference that a user might have on $(r, t(r))$:

²Arguably, the function can have other numbers of segments. For example, the segment number of the preference function in Fig. 4 can be changed to 6, where segment CD becomes shorter (260–290, flat at “fully satisfied”), assume a D' such that DD' is 290–320 (decreasing), assume an E' such that $D'E'$ is 320–325 (flat at “OK”), and $E'E$ is 325–350 (decreasing). In fact, the more segments are allowed, the smoother the function becomes, and the more easily to represent equal alternatives (i.e., assigned the same satisfaction degrees at different levels rather than only at “fully satisfaction”). However, in this paper we assume the function consists of five segments and leave the case of more segments as future work.

$$p(r, t(r)) = \begin{cases} 0 & \text{if } 200 \leq r < a_{\text{rate}}, \\ \frac{r - a_{\text{rate}}}{b_{\text{rate}} - a_{\text{rate}}} & \text{if } a_{\text{rate}} \leq r < b_{\text{rate}}, \\ 1 & \text{if } b_{\text{rate}} \leq r \leq c_{\text{rate}}, \\ \frac{d_{\text{rate}} - r}{d_{\text{rate}} - c_{\text{rate}}} & \text{if } c_{\text{rate}} < r \leq d_{\text{rate}}, \\ 0 & \text{if } d_{\text{rate}} < r \leq 380, \end{cases} \quad (1)$$

where $a_{\text{rate}}, b_{\text{rate}}, c_{\text{rate}}$, and d_{rate} are constants chosen from the range of attribute *rate*. The above preference on tradeoff alternatives can be clearly visualised (see Fig. 4 for an example). In Fig. 4, point C on the trapezoid curve corresponds to rate £260 and then from point C we can see the corresponding value of *distance* is about 20 min. That is, the tradeoff alternative corresponding to C is $(\text{rate}, \text{distance}) = (260, 20)$ which the renter is completely satisfied with (shown on the lower graph). Similarly, we can see that the tradeoff alternative at point D $(\text{rate}, \text{distance}) = (320, 15)$ also completely satisfies the user. Furthermore, the renter is completely satisfied with all the alternatives on segment CD of the tradeoff curve according to the trapezoid curve.

This method of representing preferences (as shown in the lower part of Fig. 4) was chosen because of its expressive power. Specifically, trapezoid formula (1) can model the following types of preferences (by adjusting its four parameters):

- in the case $a_{\text{rate}} = b_{\text{rate}} = c_{\text{rate}} < d_{\text{rate}}$, it models a decreasing preference on the set of tradeoff alternatives $(r, t(r))$;

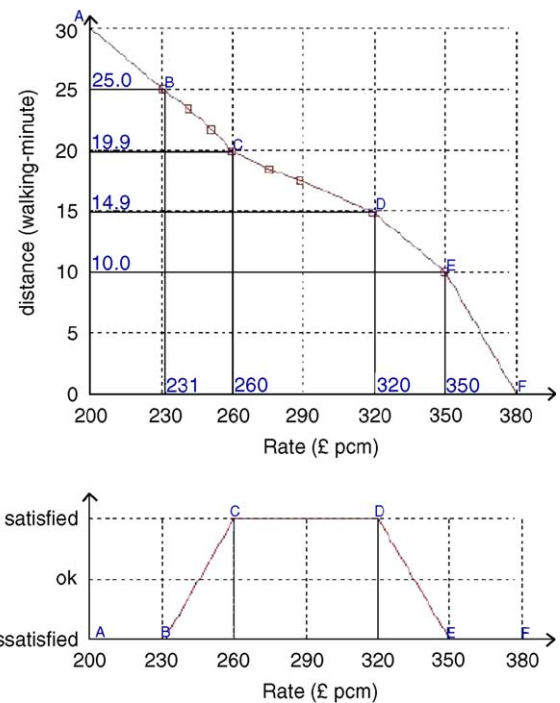


Fig. 4. The preference on tradeoff alternatives on the curve of tradeoff strategy.

- in the case $a_{\text{rate}} < b_{\text{rate}} = c_{\text{rate}} = d_{\text{rate}}$, it models an increasing preference on the set of tradeoff alternatives $(r, t(r))$;
- in the case $a_{\text{rate}} < b_{\text{rate}} = c_{\text{rate}} < d_{\text{rate}}$, it models a preference that is increasing between $(a_{\text{rate}}, t(a_{\text{rate}}))$ and $(b_{\text{rate}}, t(b_{\text{rate}}))$ but decreasing between $(c_{\text{rate}}, t(c_{\text{rate}}))$ and $(d_{\text{rate}}, t(d_{\text{rate}}))$; and
- in the case $a_{\text{rate}} < b_{\text{rate}} < c_{\text{rate}} < d_{\text{rate}}$, it models a preference that is increasing between $(a_{\text{rate}}, t(a_{\text{rate}}))$ and $(b_{\text{rate}}, t(b_{\text{rate}}))$, decreasing between $(c_{\text{rate}}, t(c_{\text{rate}}))$ and $(d_{\text{rate}}, t(d_{\text{rate}}))$, and reaching the maximum preference level between $(b_{\text{rate}}, t(b_{\text{rate}}))$ and $(c_{\text{rate}}, t(c_{\text{rate}}))$.

3. The default-then-adjust acquisition method

In the previous section, we presented the basic concepts of tradeoff strategies and preferences. Given these, we can now present our default-then-adjust method for acquiring them. The overall method is detailed in Fig. 5 and in the rest of this section we outline the main constituent components.

First, the system queries the user about choice features in order to determine which attributes to tradeoff (lines 6–10). For example, in the accommodation renting scenario, a renter (buyer) can consult the system about how to reduce the rental rate by means of a tradeoff. The system could query him about related attributes that could be used to tradeoff (e.g., take accommodation that is further away from his place of study or for a longer rental period). Then if the user indicates that he is willing to take the accommodation that is further away, the system will suggest making the tradeoff between *distance* and *rate*. If the user has no firm idea, the system can analyse with him according to its knowledge base of choice rules (see Section 4.1 for examples of such rules).

Second, in order to shape the tradeoff strategy, the system initially presents a default (lines 11–14). Then it pretends it is a seller and makes a number of concessions on one attribute and asks the user (buyer) for the limit of the other attribute he would be willing to worsen to compensate (lines 21–24).³ It then re-shapes the tradeoff strategy according to the user's adjustments (lines 16–20). Later on, the user can make more adjustments in this way (lines 16–20) or he can ask the system to set the tradeoff strategy back to its default (lines 30–34). For example, in the accommodation renting scenario, the system plays the role of an estate agency (seller) by asking the renter (buyer) the maximum distance he can accept if the rental rate is decreased from £380 to £350 pcm, and then

what if it is further reduced from £350 to £290 pcm, and then to £230 pcm. Making adjustments in this way means the space of alternatives is explored reasonably systematically and, as discussed in Section 1, is consistent with the negotiation game advocated in standard negotiation theory.

Third, the system can re-shape the default tradeoff preference according to the user's adjustment of the four parameters (see Definition 2 and formula (1)) that configure the tradeoff preference on the alternatives in the strategy set (lines 25–28). Also, at any point of the process, the user can ask the system to set the tradeoff preference back to its default (lines 32–34).

Finally, we can see that after adjusting the tradeoff preference a number of times, the user can go back to adjust the tradeoff strategy again. Also, the user can go back to any previous point of the acquisition process or even to re-start it all again (lines 35–38).

In more detail, the system consists of two knowledge bases and several functions that are explained as follows. The knowledge bases that are applied in the acquisition process are a set of *choice rules* and a set of *adjustment questions* (examples of both for our accommodation scenario are given in Sections 4.1 and 4.2). In this context, a *choice rule* is used to choose the attributes between which a potential tradeoff might exist. The condition part of a choice rule consists of some choice features that the system will ask a user to confirm or deny, while its conclusion part specifies the attributes between which the tradeoff exists. We can have such rules because various attributes between which tradeoffs exist have been identified in human negotiation theory (Steel and Beasor, 1999; Unt, 1999; Volkema, 1999). For example, in business negotiation, tradeoffs often exist between *price* and *quality*, between *price* and *quantity*, and so on. An *adjusting question* is used to help users to obtain their desired tradeoff strategy and to think about the way this strategy is structured. Such questions are of the form of how much the user (buyer) is willing to worsen one attribute to get another improved by the sellers. For example, what is the maximum distance to the university the student (buyer) is willing to accept if the estate agency (seller) decreases the rental rate from £290 to £230 pcm?

The functions that are employed in the acquisition process are as follows. (i) Function *choice-inference* (a) gives the questions of whether a component of the condition part of a choice rule holds in order to choose the attributes that a tradeoff should be made between (i.e., the conclusion part of a choice rule), and (b) chooses the attributes according to the user's replies. (ii) Function *default-strategy* plots the default strategy of the tradeoff between the chosen attributes. (iii) Function *default-preference* plots the default preference on the tradeoff alternatives in the tradeoff strategy set. (iv) Function *s-reshape* re-plots the tradeoff strategy curve (it is the default at the beginning) according to the user's adjustment to the attributes' values. That is, the segments

³In a negotiation setting, normally it is hard for the buyer (the user of the acquisition system) to reveal such information to his opponent (the seller whom the system pretends to be). However, the system is actually not the buyer's opponent, rather it is on buyer's side (being an interface to the buyer's negotiating agent). As a result, it should not be a problem for him to reveal such information to the system.

```

1. restart←‘true’; adjust-question-number←0;
2. query.content←start-question; query.performative←‘choice-question’;
3. present(query);
4. Repeat
5.   receive(response);
6.   If restart=‘true’ Then {
7.     inference-result←choice-inference(response.content, choice-rules);
8.     If inference-result∉attribute-set Then {
9.       query.content←inference-result;
10.      query.performative←‘choice-question’;}}
11.   Else {restart←‘false’; query.performative←‘default-ready’;
12.     tradeoff-strategy←default-strategy(inference-result);
13.     tradeoff-preference←default-preference(tradeoff-strategy);
14.     query.content←tradeoff-strategy+tradeoff-preference;}
15.   If response.performative∈{ready-to-adjust,reshape-strategy}Then{
16.     If response.performative=‘reshape-strategy’ Then {
17.       tradeoff-strategy←s-reshape(response.content,tradeoff-strategy);
18.       tradeoff-preference←update(tradeoff-preference,tradeoff-strategy);
19.       query.performative←‘reshaped-strategy’;
20.       query.content←tradeoff-strategy+tradeoff-preference;}
21.     If adjust-question-number≤max-adjust-question-number Then {
22.       query.content←get-adjust-question(adjust-question-set)’;
23.       query.performative←‘to-adjust-strategy’;
24.       adjust-question-number←adjust-question-number+1;}}
25.   If response.performative=‘reshape-preference’ Then {
26.     tradeoff-preference←p-reshape(response.content, tradeoff-preference);
27.     query.performative←‘reshaped-preference’;
28.     query.content←tradeoff-preference;}
29.   If response.performative∈{default-strategy,default-preference}
30.   Then {If response.performative=‘default-strategy’ Then
31.     tradeoff-strategy←default-strategy(inference-result);
32.     tradeoff-preference←default-preference(tradeoff-strategy);
33.     query.performative←‘default-ready’;
34.     query.content←tradeoff-strategy+tradeoff-preference;}
35.   If response.performative=‘restart’ Then {
36.     restart←‘true’; adjust-question-number←0;
37.     query.content←start-question;
38.     query.performative←‘choice-question’;}
39.   present(query);
40. Until response.performative=‘end’;
41. Return tradeoff-strategy+tradeoff-preference.

```

Fig. 5. The Java-like specification of the acquisition agent’s behaviour.

of the curve along the previously adjusted points are kept unchanged, but the segment that links the left and right points that are nearest to the user’s current adjusting point is replaced by two segments that link the left point to the adjusting point and then to the right point. We can do this because the tradeoff strategy is assumed to be continuous (see Definition 1). (v) Function *update* keeps the shape of the tradeoff preference but resets the updated set of tradeoff alternatives (i.e., the updated tradeoff strategy). (vi) Function *get-adjust-question* gets a new question from the adjustment question set. (vii) Function *p-reshape* replots the preference on tradeoff alternatives in the tradeoff strategy set according to the user’s adjustments of the preference configuration parameters (see Definition 2 and formula (1)). (viii) Function *present* sends the user the queries generated by the system about the attribute choice, tradeoff strategy and preference. (ix) Function *receive*

reads in the user’s responses about attribute choice, tradeoff strategy adjusting points, and the preference configuration points. It also reads in the user’s requests for setting back to the default tradeoff strategy and preference and for restarting or ending the acquisition process.

Finally, we discuss how the user communicates with the acquisition agent system. Since the human user can also be regarded as an agent (although not a software one), the communication interface between the acquisition agent and its user consists of a set of messages they send to one another. As is common in the field of agent communication (Willmott et al., 2000), we separate the representation of these messages into the *communication language* level and the *content language* level.

In the former case, we adopt the KQML standard (Finin et al., 1994) with several minor additions. Specifically, we

provide performatives through which the acquisition agent can:

- ask the user to answer a choice question (line 10),
- tell the user the default tradeoff strategy and preference are ready (lines 11 and 33) for him to adjust,
- show the user the reshaped strategy (line 19) and reshaped preference (line 27), and
- ask the user to adjust strategy (lines 23);

and through which the user can:

- tell the acquisition agent that he is ready to adjust the tradeoff strategy (line 15),
- ask the acquisition agent to reshape strategy (lines 15 and 16) and reshape preference (line 25) according to his adjustments,
- ask the acquisition agent to reset the default strategy (lines 29–30) and the default preference (line 29), and
- ask the acquisition agent to restart (line 35) or end (line 40) the whole acquisition process with the acquired tradeoff strategy and preference.

In the latter case, a query of the acquisition agent is a piece of information about a choice question (line 9), tradeoff strategy or preference (lines 14, 20, 28, and 34), or an adjustment question (line 22). A user's response is a piece of information about a reply to a choice question (line 7), a reply to an adjustment question (line 17), or an adjustment to the tradeoff strategy (line 17) or preference (line 26).

4. Prototype for an accommodation renting scenario

To demonstrate the effectiveness of our method for acquiring negotiation tradeoff strategies and preferences, we have implemented a prototype system for the accommodation renting scenario. Specifically, we consider the case in which a prospective tenant (student, the buyer) wants to rent accommodation from a real estate agent (the seller). As outlined in Section 3, the acquisition process consists of three stages: (i) choosing the attributes between which the tradeoff should be made (Section 4.1), (ii) shaping the user's strategy for tradeoffs between the chosen attributes (Section 4.2), and (iii) shaping the preferences on the tradeoff alternatives (Section 4.3). Each of these will now be dealt with in turn before we present some general remarks about the system (Section 4.4).

4.1. Choosing which attributes to tradeoff

In this stage (see Fig. 6), the system carries out a structured interview (an organised communication (Schreiber et al., 2000)) with the user. According to choice rules, the system asks the user questions and then maps the user's response information into attributes between which the tradeoff can be made. This mapping is performed in the following manner. The system puts a question in a dialogue box and after the user chooses an answer (that is the text canned in the box), the box disappears. However, the transcript of the whole conversion is recorded and displayed in order for the user to trace and analyse the choice process. The knowledge that the system uses at this stage consists of the choice rules and their explanations. In

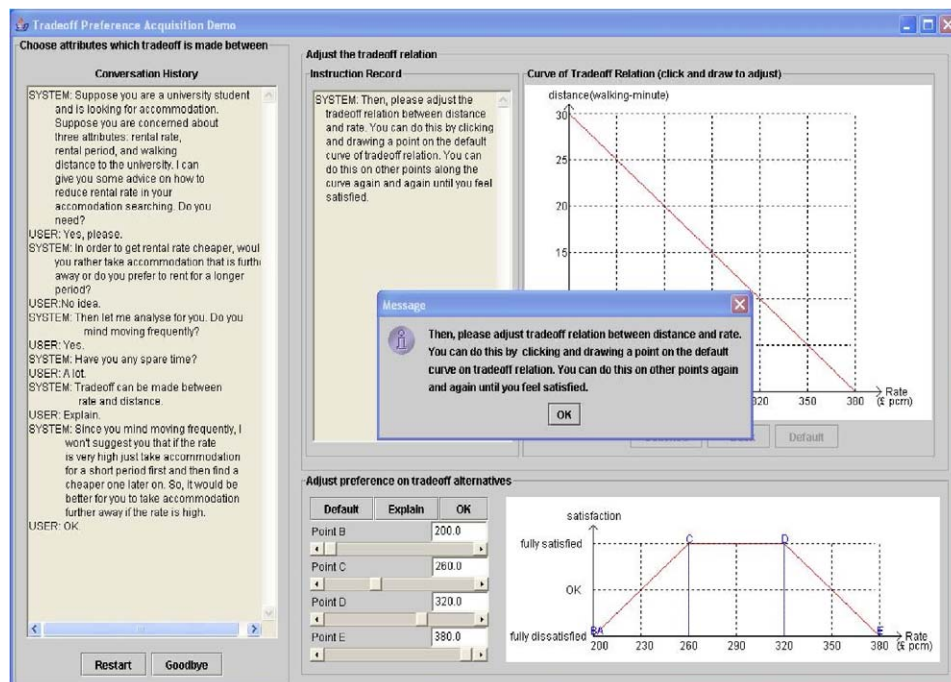


Fig. 6. Structured interview for choosing which attributes to tradeoff.

this work, the explanation facility is important because it allows the user to understand the system's advice on how to choose among the negotiation attributes. In fact, if users can understand the system's behaviour well, their trust in the system will increase. When the first stage is over, the system will direct the user to the second stage. The system also sets up a “Restart” button for the user to re-start the choice process in case he changes his mind (which he can do at any stage of the acquisition process).

The choice rules that the system uses at this stage consist of two parts: condition and conclusion. The condition part involves attributes: (1) *moving* taking value in {“mind”, “do not mind”, “neutral”} that the user is allowed to use to answer the system's enquiry about how much he minds moving, and (2) *spare-time* taking value in {“a lot”, “a little”, “some”} that the user is allowed to use to answer the system's enquiry about how busy he is. All the possible combinations of two attributes are put into consideration in these rules, and so there is only ever one rule that is applicable. The conclusion part is the suggestion that between which attributes among {*rate*, *period*, *distance*} the tradeoff should be made. More specifically, the rules are as follows:⁴

- R_1 : IF moving = “do not mind” \wedge spare-time = “some”
THEN use “tradeoff between rate and period” .
- R_2 : IF moving = “do not mind” \wedge spare-time = “a lot”
THEN use “tradeoff between rate and period” .
- R_3 : IF moving = “do not mind” \wedge spare-time = “a little”
THEN use “tradeoff between rate and distance” .
- R_4 : IF moving = “neutral” \wedge spare-time = “a lot”
THEN use “tradeoff between rate and period” .
- R_5 : IF moving = “neutral” \wedge spare-time = “some”
THEN use “tradeoff between rate and period” .
- R_6 : IF moving = “neutral” \wedge spare-time = “a little”
THEN use “tradeoff between rent and distance” .
- R_7 : IF moving = “mind”
THEN use “tradeoff between rate and distance” .

The explanations of the above rules are as follows:

Rules 1, 2, 4 and 5: Since the user does not mind moving frequently (Rules 1 and 2) or the user is neutral about moving frequently (Rules 4 and 5), if the rental rate is very high he would prefer renting the accommodation for a short period first while he endeavours to find a cheaper

one. Although this may take a considerable amount of time, he can do this because he has some (Rules 1 and 5) or a lot (Rules 2 and 4).

Rules 3 and 6: Since the user is neutral about moving frequently, if the rental rate is very high, it should be fine for him to take the accommodation for a short period first and find a cheaper one later on. However, this may take a considerable amount of time and this is problematic (since he has only a little spare time). So, it would be better for him to reduce the rental rate by taking accommodation that is further away.

Rule 7: Since the user objects to moving frequently, it is not a good idea to suggest to him to take the expensive accommodation for a short period first and then move to a cheaper one later. Rather, it is better for him to take the accommodation further away to get the rental rate down.

4.2. Shaping the tradeoff strategy

After the stage of choosing the tradeoff attributes has been completed, the system moves the user onto the stage of determining the tradeoff strategy (see Fig. 7). To this end, the system first asks the user a number questions to help him adjust the points along the tradeoff strategy curve (e.g., adjust point (*distance*, *rate*) = (5, 350) to (*distance*, *rate*) = (13.9, 350), see Fig. 7). These questions are: what is the maximum distance or period he can accept if the rental rate is decreased from £380 to £350 pcm, and then what if it is further reduced from £350 to £290 pcm, and then to £230 pcm? Then, the user can further adjust any other points he choose simply by clicking on the relevant points and dragging them to the desired positions. If the user is satisfied with the curve shape of the tradeoff strategy, he can press button “Satisfied” and the system directs him to the next stage (see Fig. 8); otherwise he can continue to adjust the curve until he is satisfied. During the adjustment process, the user can press button “Back” to retrieve the previous adjustment, button “Default” to go back to the original default tradeoff strategy,⁵ button “Restart” to restart the whole process.

4.3. Shaping the tradeoff preference

After the stage of shaping the tradeoff strategy is over, the system moves onto shaping the tradeoff preference (see Fig. 9). Thus, the system shapes the user's preference on

⁴These rules could be acquired for a given domain by using some relatively standard knowledge discovery methods. For example, the rough set theory as detailed in Pawlak (1982, 1991) provides a theoretic basis for discovering decision rules from sample data. Here the choice rules are a kind of decision rules since they indicate under which condition which attributes the users should tradeoff between. Thus, rough set theory can be employed to generate such choice rules. However, this is beyond the scope of this paper and in this case they were developed manually based on domain expertise.

⁵The default strategy is fixed to be a decreasing line meaning that the degree to which one attribute is worsened is the same as the degree to which another attribute is improved (see Fig. 3). In this case, the two attributes are assumed to be equally important. If the user does not think so, he can adjust to a concave or a convex curve (see Figs. 1 and 2). If a concave or a convex curve was chosen as the default strategy, intuitively the user might feel confused: why did the system think I had viewed one particular attribute being more important than another at the beginning of adjustment process when it has no information about my tastes? Thus, the decreasing line is chosen since it assumes the user is neutral about the attributes under consideration.

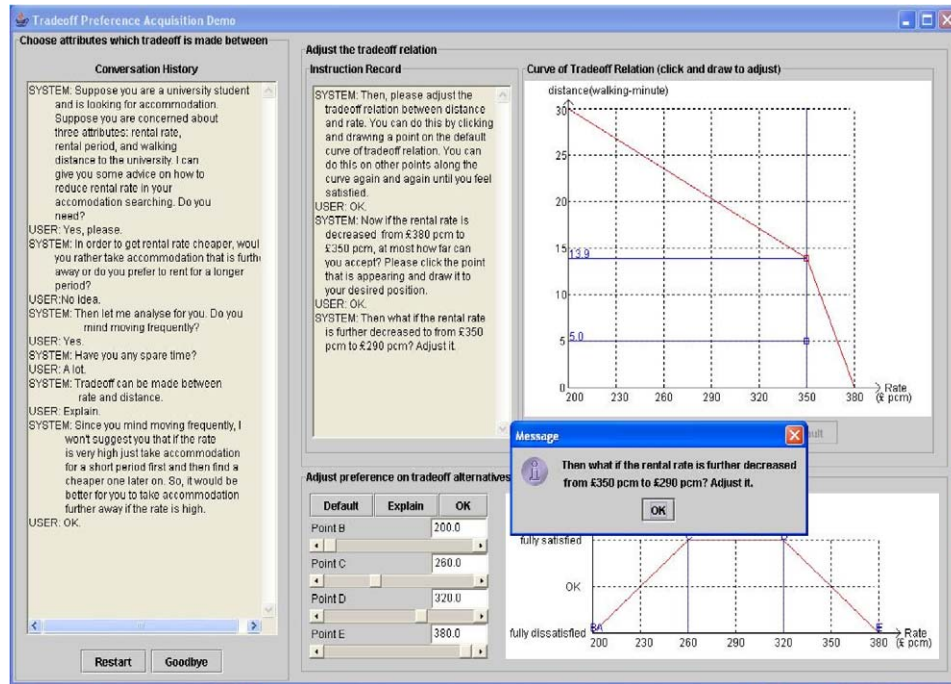


Fig. 7. Shaping the tradeoff strategy by clicking and drawing some points along the curve.

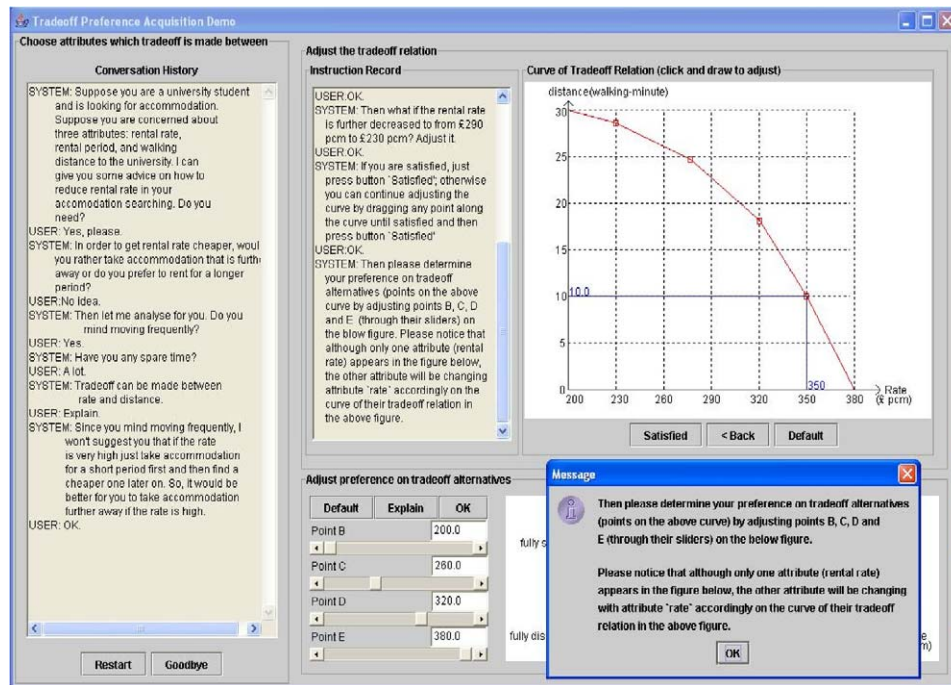


Fig. 8. Displaying the tradeoff strategy and directing the user to the stage of shaping the tradeoff preference.

tradeoff alternatives according to his adjustment of the tradeoff preference parameters (see Definition 2). This can be performed by adjusting the sliders of points *B*, *C*, *D* and *E* or filling in their values in the corresponding boxes.

If the user does not fully understand the meaning of the preference curve, the “Explain” button gives the following extra details:

Segment AB: Tradeoff alternatives in this segment are unsuitable. This is because the attribute (e.g., period or distance) that is worsened (to improve the other) is beyond the maximum limit the user is willing to accept (even though the other (e.g., rate) can be made even better).

Segment BC: Tradeoff alternatives in this segment are preferred increasingly. This is because the attribute

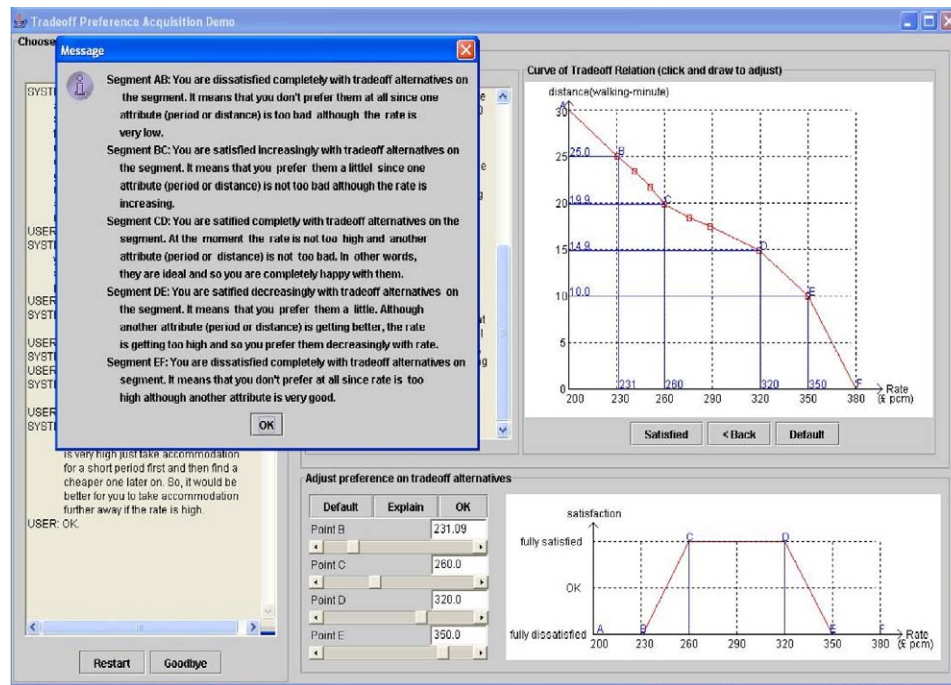


Fig. 9. Adjusting preference parameters and explaining their meaning.

(e.g., period or distance) that is worsened (to improve the other) is becoming increasingly acceptable although the other attribute (e.g., rate) is worsening (a little).

Segment CD: Tradeoff alternatives in this segment are the most preferred (completely satisfactory). At this point, for example, the rate is not too high and the other attribute (period or distance) is also quite satisfactory.

Segment DE: Tradeoff alternatives in this segment are preferred decreasingly when the attribute (e.g., rate) approaches the maximum limit the user is willing to accept. This is because the attribute (e.g., rate) to be traded off is becoming increasingly unacceptable although the other attribute (period or distance) is improving.

Segment EF: Tradeoff alternatives in this segment are absolutely unacceptable because the attribute (e.g., rate) is beyond the user's maximum limit (e.g., the user's budget) even though the other attribute might be very good.

4.4. General remarks

Having outlined our system and how it operates, we first evaluate it from a qualitative perspective (the quantitative evaluation is given in Section 5). Now according to [Turban and Aronson \(1998\)](#), there are a number of objectives that a knowledge acquisition system should possess. These include:

- *The ability to acquire knowledge directly from the user (not through a human knowledge engineer).* The system developed in this paper functions as an experienced knowledge engineer that interviews the human users directly in order to obtain information about their tradeoff strategies and preferences.

- *A tutorial ability.* Generally speaking, the capability of explanation for behaviour is an essential component in knowledge based systems ([Schreiber et al., 2000](#)). For this reason, we set up the “Explain” button so the users can check the rationale of the system's advice about the choice of tradeoff attributes and the shape of the tradeoff strategies and preferences. Such explanations help the users to use the system efficiently and to trust the system's advice.
- *The ability to analyse work in progress so that users can detect inconsistencies and gaps (if any) in knowledge.*⁶ The explanation facility, the interview record, the system's instruction record, and the graphical user interface serve this purpose. This is because these facilities enable the user to trace and analyse the information provided to the system and to recap the way in which the system responded.
- *A user friendly interface.* A key component here is the ability to interact flexibly with users. Accordingly, we designed the system as an autonomous agent that provides the required flexibility by directing its users through the various steps of the acquisition process and by allowing the users to go back to any previous points at any time. More specifically, users can re-press any previous buttons and re-adjust the previous tradeoff

⁶In our future work, we plan to add a functionality of automatic inconsistency checking and adjusting. Currently, in the stages of acquiring tradeoff strategies and preferences there is an assumption that the result of adjustment process is roughly consistent with the users' tradeoff strategies and preferences. Our technique does not actually ask the user for a consistency check at each point in the graphs of tradeoff strategy and preference, but that consistency is already implied through the user's visual inspection of graphs.

strategy and preference, and then the system reacts accordingly.

5. Empirical evaluation

This section presents our empirical evaluation of our prototype system. Generally speaking, there is a lack of well-established evaluation approaches in the development and assessment of knowledge acquisition techniques (Shadbolt et al., 1999). So, we cannot carry out our evaluation experiment according to a standard approach. On the other hand, the success criteria should clearly reflect end-user concerns (Buchanan and Shortliffe, 1984; Gaschnig et al., 1983). Thus, we focus on investigating the following properties:

- (i) whether it is easy for the users to understand and master the system,
- (ii) whether the system is helpful for the users to specify their tradeoff strategies and preferences, and
- (iii) whether the system is efficient in accurately acquiring tradeoff strategies and preferences.

To measure the extent to which the first two properties are achieved, we can ask the users for their subjective opinions after they use the system (Section 5.1). The final property can be measured by taking into account the number of interactions that occurred between the system and the users before the user was happy with the results that were reached (Section 5.2).

In order to determine the number of user samples necessary to do a comprehensive analysis, we borrow on the principle advocated by Menzies (1998):

Sample size (N) should be carefully controlled. Small sample sizes are hard to analyse. However, as random sample sizes get larger, they approach a bell shape (the normal distribution) which is a well-understood distribution. In practise, N greater than 20 is acceptable and N greater than 30 is encouraged. On the other hand, there may be no benefit with making N very large (Cohen also argues that sample sizes of N greater than 50 can be pointless (Cohen, 1995, p. 116)).

Accordingly, we have chosen a sample size of 40 users in our experiments. They are undergraduates, master students, Ph.D. students and research fellows in both computer science and other disciplines (the number of computer scientists in the population was about 45%). Among these, eight tried earlier versions of the system and provided their feedback that was used to get the system into the present form; while the remainder used the system for the first time. For the first time users, before they evaluated the system, they were given some simple documentation about the system and how it works, and they were also given a brief tutorial (lasting about 5 min).

5.1. Subjective opinions

After the users have used our system, we ask them whether they *agree*, *neither agree nor disagree*, or *disagree* for the following statements:

- *Clear*: the system's organisation is clear.
- *Reliable*: the system's implementation is reliable.
- *Friendly*: the system's behaviour is easy to understand and follow.
- *Easy to use*: the system is easy to operate.
- *Convincing*: the system's explanation of its decisions and components is clear and convincing.
- *Professional*: the system's advice is professional.
- *Helpful*: without the system's help, it would not have been easy to articulate tradeoff strategies and preferences.
- *Like it*: overall rating of the system.

Fig. 10 shows the summary of the users' responses to the above criteria. From this, we can clearly see that for most of the users, our system is helpful in specifying their tradeoff strategies and preferences and it is easy for them to understand and master our system. Since most users are first timers and the introduction to the system and its operation is comparatively brief, it is perhaps not surprising that the scores for the *Clear*, *Friendly* and *Convincing* criteria are a bit lower than those for some of the other criteria since these are the thing that usually come with time. Indeed for the non-first time users the average values for these criteria were, respectively, about 87%, 85% and 75% that are higher than the corresponding figures for the first timers.

5.2. Accuracy versus efficiency

Now we turn our focus to discussing the evaluation of our system in terms of its accuracy with respect to the amount of information it has elicited. Of course, if the user was asked for everything, then we could assume the acquired result is accurate. However, this is impossible in

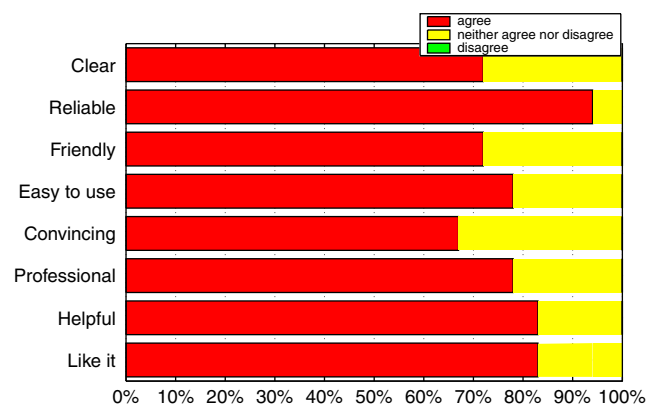


Fig. 10. Users' satisfaction with the tradeoff acquisition system.

practise because there are infinitely many possible combinations of the values of two attributes that take values on the continuous intervals. Thus, in this evaluation, we aim to determine *how much* information the system needs to have a good enough (from the user's perspective) view of his tradeoff strategies and preferences.

First of all, since different users have different tradeoff strategies and preferences (this is simply commonsense), there does not exist a standard tradeoff strategy and a set of preferences (if this were not true, the whole acquisition process would become unnecessary). Thus, it is impossible to measure the accuracy of the acquired tradeoff strategies and preferences against standard ones.

Given this, the next step is to determine whether we can measure the accuracy to which the acquired tradeoff strategies and preferences are consistent with users' orderings of the accommodation instances. There are also some problems here. Suppose in the evaluation experiment, we ask users to specify a preference ordering on a number of accommodation instances or simply put these instances into piles of likes, dislikes, and neutral. Experience dictates that it is a kind of overall preference ordering that takes into account *all* the attributes of accommodation (e.g., distance, rental period, rental rate, location, and so on) (Mas-Collell et al., 1995). Therefore, it is not appropriate to compare the user's specified instance preference ordering with the one that is implied by the acquired tradeoff preference only. This is because just *some* (e.g., rental rate and distance only, not all of them) of the attributes are considered in the acquired tradeoff preferences. Thus, in order to obtain the instance preference ordering that the acquired preference implies, we have to aggregate *not only* the tradeoff preference, *but also* the preference on all other attributes. However, there are many aggregation operators that could be used to do this, each of which will result in different ordering on users' instances (Dubois and Prade, 2004), and none of which are clearly dominant for this problem.

In the case that the user is simply required to sort a number of accommodation instances into the piles (instead of a complete preference ordering), even more problems arise. First, somebody who has little in the way of domain knowledge (e.g., an international student who has just arrived in the UK) may be unable to make informed choices. Second, the terms *like*, *dislike* and *neutral* are typical fuzzy concepts. So, after aggregating individual attributes' assessments of an accommodation, we need to map the overall assessment into one of the fuzzy piles of *like*, *dislike* and *neutral* in order to check whether the user's direct sorting of accommodation instances is consistent with that which follows from the acquired preferences. Here two more problems arise: (i) how to define the fuzzy sets of *like*, *dislike* and *neutral*, and (ii) how to map an accommodation's overall assessment (according to all the individual preferences) into one of these fuzzy sets. Now, neither of these have a standard answer and the various options all provide different overall outcomes (Dubois and Prade, 2000).

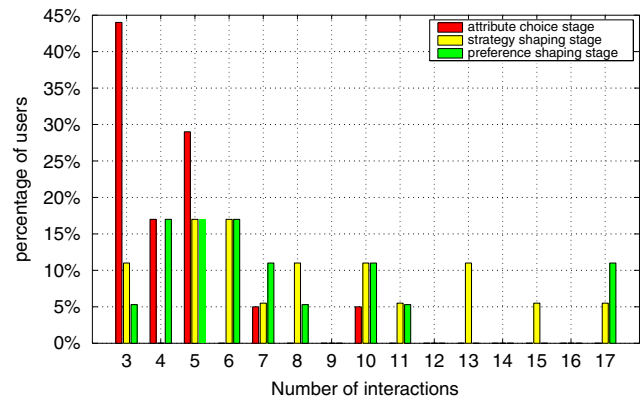


Fig. 11. Interaction numbers when satisfactory results are obtained.

From this, it should be obvious that it is extremely difficult to find an objective way of measuring the accuracy of the acquired tradeoff preferences. Thus, we consider subjective ways. Intuitively, if a user is satisfied with the tradeoff strategies and preferences that the system acquired, it is difficult to say that the accuracy of our acquisition system is not good. Thus, when a user is satisfied with the acquired result, we say the system has accurately acquired the user's tradeoff strategies and preferences.

Now only one problem remains. That is, how do we measure the amount of information the user has had to put into the system? To this end, since the more interactions there are between the system and the user, the more information the user provides, the number of interactions is indicative of the amount of information the user provides during the course of the acquisition process. Here, the lower the number of interactions, the more efficient the system. Thus, we measure the number of interactions that happen before the user is satisfied with the tradeoff strategy and preference.

In more detail, Fig. 11 shows the result of the number of interactions between users and our acquisition system.⁷ From this, we can clearly see that large numbers of interactions are not required at any stage of the acquisition process. In fact, at the attribute choice stage, the average number of interactions is 4.33. At the tradeoff strategy shaping stage, the average number is 8.39; more than 30% people interact five or six times and less than 15% people do it 15 or 17 times. And at the tradeoff preference shaping

⁷In the attribute choice stage, one interaction is when the system asks the user a question and then he replies, when the user presses an "Explain" button and the system explains its advice, or when the user presses the "Restart" button. In the strategy shaping stage, an interaction is when the system asks the user an adjustment question and they respond by clicking a point along the current strategy curve and dragging it to the desired position, when the user directly clicks a point along the current strategy curve and drags it to the desired position, or when the user presses button "Satisfied", "Back", or "Default". In the preference shaping stage, an interaction is an adjustment that the user performs on one of the parameters that configure the preference curve, or when they press one of the "Default", "Explain", and "OK" buttons.

stage, the average number is 6.39; less than 20% people interact more than 10 times. Interestingly, the users who made high numbers of interactions with the systems often had a non-computer science background. We believe they made such frequent changes because their backgrounds made them hesitant in operating the system and thus they cannot focus themselves on thinking about their tradeoff strategies and preferences. Generally speaking, however, these results show that our default-then-adjust method is efficient.

Now, in order to get a clearer idea about how good our acquisition method is in terms of efficiency, we compare it with a couple of alternatives. First, we consider the simple approach of constructing tradeoff alternatives by iterating over all possible value combinations of attributes (Keeney and Raiffa, 1983). Considering the situation of two attributes as an example, suppose they have n and m possible values, respectively. Then the total number of all possible combinations is $n \times m$. Clearly, only in the case that the space of all possible combinations is *small* enough, is it practical to require the user to give an ordering of all the combinations. Actually, in our method, the average number of adjustments is $8.39 + 6.39 = 14.78$ (the attribute choice stage is not involved). Whereas in the simple approach, as long as each attribute has more than four possible values, the times that its users need to provide information is more than $4 \times 4 = 16$. For example, in our accommodation renting domain (Luo et al., 2004), suppose attribute *distance* has 11 possible values 0, 3, 6, ..., 30, and attribute *rental rate* also has 11 possible values 0, (0, 250], (250, 260], (260, 270], ..., (320, 330], (330, ∞). Then, the times that its users need to provide information is more than $11 \times 11 = 121$. In other words, it is much less efficient than ours (considering in our approach the possible combinations are *infinite*, rather than *finite*).

Next, we consider the exhaustive method (Luo et al., 2004). In the exhaustive method, the value combination of two attributes in a tradeoff strategy must obey the property that if one attribute's value is worsened, then another's must be improved. Thus, for the best value of the attribute with n possible values, the number of the possible combinations with another attribute with m possible values is m ; for its second best value, the number is $m - 1$; ...; for its worse value, the number is $m - n + 1$ (assuming $m > n - 1$). So, the total number of tradeoff combinations is:

$$m + (m - 1) + \dots + (m - n + 1) = m \times n - \frac{(n - 1)n}{2} \leq m \times n.$$

The above analysis shows that although the idea of the exhaustive method is to systematically explore the space of possible combinations that obey the tradeoff property above, this does not mean the user has to check all possible combinations of attributes (as per the simple approach). Therefore, the exhaustive method is more efficient than the simple one. However, it is clear that the method proposed in this paper is still much more efficient than the exhaustive

method. In fact, in the exhaustive method, as long as each attribute has more than six possible values, the times its users need to provide information is more than $6 \times 6 - (6 - 1)6/2 = 21$, which is larger than that of our approach (which was 14.78). Taking the above figures for our accommodation renting domain, then the amount of times that its users need to provide information is $11 \times 11 - (11 - 1)11/2 = 66$. This is clearly less than the 121 of the simple method but far less than the average figure of 14.78 of our method.

6. Related work

The background of our work is knowledge acquisition and preference elicitation. Generally speaking, there are many methods for acquiring knowledge from the users of knowledge-based systems (see Hoffman and Shadbolt, 1995 for an overview). However, usually such methods do not aim to elicit user strategies and preferences for decision making (particularly for dynamic decision making processes such as negotiation). There is, however, some work in this area. The most common methods elicit a user's preferences by means of utility functions. For instance, in gambling elicitation techniques (Neumann and Morgenstern, 1944) the utility of one outcome is ascertained by comparing it to a lottery involving two others, and in visual exploration and incremental utility elicitation (Blythe, 2002) suggestions are made visually to a human user based on an incomplete model of their utility function and the model is updated based on the received feedback. However, the workload of users in using such methods can be very high.

In order to reduce this workload, a number of heuristic methods have been proposed. For example, case based preference elicitation (Ha and Haddawy, 1998, 2003; Ha et al., 2001) requires a user to provide partial information about their preference and then the system constructs the whole preference structure by choosing the most similar one in the case base of preference structure. In contrast, preference elicitation via theory refinement (Geisler et al., 2001; Restificar et al., 2002; Haddawy et al., 2003) starts with an approximate and incomplete body of domain knowledge and then corrects inaccuracies and incompleteness by training on examples. However, such preference elicitation methods are also unsuitable in this context (although a procedure of negotiation can certainly be viewed as one of dynamic decision making over time). This is because in the case of the preference elicitation for *ordinary* decision making (Ha et al., 2001; Ha and Haddawy, 2003), the term *preference* refers to *orderings* on a set of *alternatives*, and the focus is basically on eliciting the *ordering* (assuming the alternatives are known or given). In contrast, in this paper, our focus is on actually eliciting the user's tradeoff *alternatives*. As to the problem of eliciting the *ordering* on the set of tradeoff *alternatives*, we use an adjusting parameter method (thus, in future work, we can take advantage of general preference *ordering*

elicitation techniques in our method for eliciting tradeoff strategies and preferences).

Tradeoff strategies and preferences can also be viewed as a form of constraint (i.e., such a constraint is the set of the value combinations of attributes between which the tradeoff is made and each combination is assigned a preference level) (Luo et al., 2003b; Perloff, 1998). However, while existing techniques for constraint elicitation have shown some success in particular contexts, they are not easily able to handle our tradeoff acquisition problem. For example, in O'Connell et al. (2002), O'Connell, O'Sullivan and Freuder present an approach to interactive constraint acquisition based on the techniques of Mitchell's *List-Then-Eliminate* method (Mitchell, 1997). In this approach, there is a "hypothesis space" of constraints over which a general-to-specific ordering is known (or is implicit in the hypothesis space representation). Then the user's examples (where a constraint should be met or violated) are used to eliminate hypotheses that are inconsistent with the examples. In this procedure, a strategy is also employed to minimise the dialogue length between the user and the computer system. However, this method is only applicable to crisp constraints, but buyers' tradeoff strategies and preferences are often fuzzy constraints (Luo et al., 2003b). In Biso et al. (2000), Biso, Rossi and Sperduti used neural networks to tackle the problem of soft constraint acquisition (fuzzy constraints are a special case of soft constraints). However, their method is used to learn the solution rating function given a fixed constraint graph and some examples of solution ratings. Thus, their method assumes the constraints are already available, and what their method learns is just how to rate a solution according to the available constraints. It does not resolve the problem of how to acquire soft constraints themselves (in particular, the fuzzy constraints of buyers' tradeoff strategies and preferences).

Probably the work that is most related to ours is Freuder and O'Sullivan (2001). They aim to acquire tradeoffs between the attributes that configure an object. However, there are a number of important differences. Firstly, their method uses heuristic strategies to generate various tradeoffs and then asks the user to confirm whether they are acceptable (their aim is to improve the efficiency of the acquisition process). This approach may cause some tradeoffs to be missed in complex multi-dimensional spaces and so we put the whole space of tradeoff alternatives into consideration. Secondly, their method is not designed for negotiation problems, and, their example scenario, the N-Queens problem, is much more objective (i.e., independent of users' individual opinions) than that typically found in negotiations. Finally, their method assumes that each tradeoff alternative is equivalent (i.e., users have no preference on alternatives), whereas we assume that for different tradeoff alternatives the user has different satisfaction degrees (i.e., the user has a preference on tradeoff alternatives).

Turning now to the area of agent-based automated negotiation. As noted previously, little work has been done to address the acquisition problem of the strategies and preferences of the human users for whom the negotiating agents are acting. Nevertheless:

- In Hudson and Sandholm (2002), Restificar and Haddawy (2003), different methods have been developed for one agent to elicit preferences from another agent in combinatorial auction and bilateral negotiation settings, respectively. However, they both assume that the users of the agents have already imparted their preferences into the negotiating agents (the aim of our work presented in this paper). Thus, the aim of the elicitation process in Hudson and Sandholm (2002) is to prune the auctioneer agent's search space; and in Restificar and Haddawy (2003) the negotiating partner's preference models are used to predict its behaviour during the course of negotiation.
- In Boutilier et al. (2003), negotiation and preference elicitation are linked. However, this work differs from ours in the following aspects. (i) Their preference elicitation is not from *human users* but from *programs* (i.e., an automated resource manager program elicits the preferences on resources from workload manager programs). In contrast, the aim of our work is to elicit tradeoff strategies and preferences directly from automated negotiation programs' *users*. (ii) Their preference elicitation process is a cooperative negotiation between an automated resource manager and the workload managers. In other words, the motivation of the elicitation is not for a program to negotiate faithfully on behalf of its human owner (the motivation of our work). Rather, the motivation of Boutilier et al. (2003) is to find near-optimal allocations of resources to distinct computing elements in large, distributed computing systems.
- In previous work (Castro-Schez et al., 2004), we used fuzzy repertory tables mainly for acquiring the rewards and restrictions that a seller has on specific products for a specific negotiation model that we have developed (Luo et al., 2003b). However, this method does not deal with a buyer's tradeoff strategies nor preferences.
- The work of Guo et al. (2003) employs an evolutionary algorithm to address the problem of preference elicitation (in the form of utility functions) for multi-attribute negotiation. However, their term *preference* does not particularly refer to tradeoff preference. As a result, Guo et al. (2003) cannot easily be distinguished from the general research area of user preference elicitation.

Finally, by way of adjusting the default curves for tradeoff strategies and preferences, the method developed in this paper removed the limitations of our two previous methods—the exhaustive (Luo et al., 2004) and the knowledge-based method (Luo et al., 2003a), while keeping their advantages. In fact, in the former case, systematically exploring the space of all possible tradeoff alternatives

becomes very expensive if the sizes of the attribute domains are large (as shown in Section 5.2). Thus, in this case, for practical purposes, one cannot assume the user will answer all the required comparisons willingly and accurately. To remove this limitation, the acquisition method proposed in this paper reduces the user's workload by offering defaults to kick start the process, and then lets the user do any necessary adjustments until he is satisfied. Moreover, our empirical evaluation confirms that most users actually perform a small number of adjustments before satisfactory results are obtained. On the other hand, our system keeps the flexibility of the exhaustive method that users can specify any shape of tradeoff strategy and preference they wish. In the latter case, the knowledge-based method reduced the workload by restricting the users to only make a choice among a number of fixed tradeoff strategies. However, this lacks the flexibility of allowing the user to adjust the defaults presented to them.

7. Conclusions and future work

In order for autonomous agents to negotiate faithfully on behalf of their human owners, it is critical to elicit tradeoff strategies and preferences from their owners. We believe such knowledge is fundamental to many negotiation models, but, to date, it is an area that has been largely neglected. To this end, we developed a default-then-adjust knowledge acquisition method and an accompanying software tool to implement this method. Our method is consistent with the principles of standard negotiation theory and can acquire tradeoff strategies and preferences between pairs of attributes. Our evaluation of the method and the tool indicates that they are effective, flexible, and efficient in their task.

However, for the future more work is still needed. Specifically:

- The method needs to be examined across a wide range of domains in order to ensure that it is sufficiently generic. We believe this is the case since there is nothing specific to the accommodation renting domain in our model,⁸ but this hypothesis requires validation.
- There is no objective measure or gold standard against which the acquired model can be evaluated. Thus, the success of a given acquisition model can only be determined by coupling the acquired tradeoff strategies and preferences to various negotiation models and evaluating the users' satisfaction on the outcomes. However, even in such cases there is the problem of determining whether any discontent is due to the behaviour of the negotiation models or whether the user simply did not fully appreciate the ramifications of their tradeoff strategies and preferences.
- We believe that we can learn from work in the general area of preference elicitation where many heuristic methods have been used to reduce the elicitation workload, and so we wish to explore the use and the effectiveness of methods other than default-then-adjust. In addition, our preference ordering function is assumed to be a (potentially) piecewise nonmonotonic function consisting of five segments. However, the more segments that are allowed, the smoother the function becomes, and the easier it is to represent equal alternatives (i.e., to assign the same satisfaction degree at different levels rather than only at “fully satisfaction”). Thus, it is worth investigating the increased expressiveness that more segments permit.
- The acquisition of other types of tradeoffs needs further consideration. Such tradeoffs would deal with (i) more than pairwise considerations, (ii) *non-linear* preferences⁹ (e.g., a user may prefer tradeoff alternative *a* to tradeoff alternative *b*, and tradeoff alternative *a* to alternative *c*, but have no preference ordering between *b* and *c*), (iii) fuzzy curves for tradeoff strategy (e.g., if rate is *about* £260 pcm then the distance is *about* 20 min walk and if rate is *about* £270 pcm then the distance is *about* 15 min walk), and (iv) the reality of a breakdown or an impasse in a negotiation.

Acknowledgements

The authors would like to thank the anonymous referees for their insightful comments that have significantly improved the quality of the paper. The work is supported by Hewlett Packard Research Labs. The authors are also obliged to Partha S. Dutta for his proof-reading of an earlier version of this article, to Minghua He for her help in programming the software, and to the people who participated in our evaluation experiment for their feedback.

References

- Beliakov, G., Warren, J., 2001. Appropriate choice of aggregation operators in fuzzy decision support systems. *IEEE Trans. Fuzzy Systems* 9 (6), 773–784.
- Biso, A., Rossi, F., Sperduti, A., 2000. Experimental results on learning soft constraints. In: Cohn, A.G., Giunchiglia, F., Selman, B. (Eds.), *KR2000: Principles of Knowledge Representation and Reasoning*, San Francisco. Morgan Kaufmann, Los Altos, CA, pp. 435–444.
- Blythe, J., 2002. Visual exploration and incremental utility elicitation. In: *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 526–532.
- Boutilier, C., Das, R., Kephart, J.O., Tesauro, G., Walsh, W.E., 2003. Cooperative negotiation in autonomic systems using incremental utility elicitation. In: *Proceedings of the Nineteenth International*

⁸From Fig. 5, we can clearly see that the method is irrelevant to the domain knowledge of choice rules, default tradeoff strategies and preferences in the accommodation renting scenario.

⁹In the system that we proposed in this paper, a preference ordering of all tradeoff alternatives is determined by the user's satisfaction degrees on the alternatives. Since such satisfaction degrees take values on [0,1], clearly the preference they induce is a linear ordering rather than a general partial ordering.

- Conference on Uncertainty in Artificial Intelligence, August, Acapulco, Mexico, pp. 89–97.
- Buchanan, B., Shortliffe, E., 1984. Rule-based expert systems: the EMYCIN experiments of the Stanford heuristic programming project, chapter 10. Uncertainty and Evidential Support, Addison-Wesley, Reading, MA, pp. 209–232.
- Castro-Schez, J.J., Jennings, N.R., Luo, X., Shadbolt, N.R., 2004. Acquiring domain knowledge for negotiating agents: a case of study. *Internat. J. Human Computer Studies* 61 (1), 3–31.
- Cheung, S.C., Hung, P.C.K., Chiu, D.K.W., 2003. On e-negotiation of unmatched logrolling views. In: *Proceedings of the Thirty-Sixth Hawaii International Conference on System Sciences*, May, Hawaii, USA.
- Cohen, P., 1995. *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA.
- Collin, P.H., 2001. *Dictionary of Business*, third ed. Peter Collin Publishing, London.
- Dubois, D., Prade, H. (Eds.), 2000. *Fundamentals of Fuzzy Sets*. Kluwer Academic, Dordrecht.
- Dubois, D., Prade, H., 2004. On the use of aggregation operations in information fusion processes. *Fuzzy Sets and Systems* 142, 143–161.
- Faratin, P., Sierra, C., Jennings, N.R., 2002. Using similarity criteria to make issue tradeoffs in automated negotiations. *Artificial Intelligence* 142 (2), 205–237.
- Finin, T., Fritzson, R., McKay, D., McEntire, R., 1994. KQML as an agent communication language. In: *Proceedings of the Third International Conference on Information and Knowledge Management*, pp. 456–463.
- Fisher, R., Ury, W., 1981. *Getting to Yes: Negotiating an Agreement without giving in*. Random House Business Books, New York.
- Freuder, E.C., O'Sullivan, B., 2001. Generating tradeoffs for constraint-based configuration. In: *Proceedings of the Seventh International Conference on Principles and Practice of Constraint Programming*, pp. 590–594.
- Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E., Terry, A., 1983. Evaluation of Expert Systems: Issues and Case Studies. Addison-Wesley, Reading, MA, pp. 241–280 (Chapter 8).
- Geisler, B., Ha, V., Haddawy, P., 2001. User modeling via theory refinement. In: *Proceedings of the International Conference on Intelligent User Interface*, pp. 87–90.
- Guo, Y., Muller, J.P., Weinhardt, C., 2003. Learning user preferences for multi-attribute negotiation: an evolutionary approach. In: Muller, J., Marik, V., Pechoucek, M. (Eds.), *Multi-Agent Systems and Applications III*, Lecture Notes in Artificial Intelligence, vol. 2691. Springer, Berlin, pp. 303–313.
- Ha, V., Haddawy, P., 1998. Toward case-based preference elicitation: similarity measures on preference structures. In: *Proceedings of the Fourteenth International Conference on Uncertainty in Artificial Intelligence*, pp. 193–201.
- Ha, V., Haddawy, P., 2003. Similarity of personal preferences: theoretical foundations and empirical analysis. *Artificial Intelligence* 146 (2), 149–173.
- Ha, V., Haddawy, P., Miyamoto, J., 2001. Similarity measures on preference structure, Part II: utility functions. In: *Proceedings of the Seventeenth International Conference on Uncertainty in Artificial Intelligence*, pp. 186–193.
- Haddawy, P., Ha, V., Restificar, A., Geisler, B., Miyamoto, J., 2003. Preference elicitation via theory refinement. *J. Machine Learning Research* 4, 317–337 Special Issue on Fusion of Domain Knowledge with Data for Decision Support.
- He, M., Jennings, N.R., Leung, H.F., 2003. On agent-mediated electronic commerce. *IEEE Trans. on Knowledge and Data Engineering* 15 (4), 985–1003.
- Hoffman, R.R., Shadbolt, N.R., 1995. Eliciting knowledge from experts: a methodological analysis. *Organizational and Human Decision Process* 62 (2), 129–158.
- Hudson, B., Sandholm, T., 2002. Effectiveness of preference elicitation in combinatorial auctions. In: Padget, J., Shehory, O., Parkes, D., Sadeh, N., Walsh, W.E. (Eds.), *Agent-Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, Lecture Notes in Computer Science, vol. 2531, pp. 69–86.
- Keeney, R., Raiffa, H., 1983. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, Cambridge.
- Lomuscio, A.R., Wooldridge, M., Jennings, N.R., 2003. A classification scheme for negotiation in electronic commerce. *Internat. J. Decision and Negotiation* 12 (1), 31–56.
- Luo, X., Zhang, C., 1999. An axiom foundation for uncertain reasonings in rule-based expert systems: NT-algebra. *Knowledge and Inform. Syst.: An Internat. J.* 1 (4), 415–433.
- Luo, X., Jennings, N.R., Shadbolt, N., 2003a. Knowledge-based acquisition of tradeoff preferences for negotiating agents. In: *Proceedings of the Fifth International Conference on Electronic Commerce*, Pittsburgh, USA, pp. 138–144.
- Luo, X., Jennings, N.R., Shadbolt, N., Leung, H.F., Lee, J.H.M., 2003b. A fuzzy constraint based model for bilateral, multi-issue negotiation in semi-competitive environments. *Artificial Intelligence* 148 (1–2), 53–102.
- Luo, X., Jennings, N.R., Shadbolt, N., 2004. Acquiring tradeoff preferences for automated negotiations: a case study. In: *Agent-Mediated Electronic Commerce V: Designing Mechanisms and Systems*, Lecture Notes in Computer Science, vol. 3048, pp. 37–55.
- Mas-Collell, A., Whinston, M.D., Green, J.R., 1995. *Microeconomic Theory*. Oxford University Press, Oxford.
- Menzies, T., 1998. Evaluation issues for problem solving methods. In: *Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management*.
- Mitchell, T., 1997. Concept learning and the general-to-specific ordering. In: *Machine Learning*, McGraw Hill, New York, pp. 20–51 (Chapter 2).
- Mudgal, C., Vassileva, J., 2000. Bilateral negotiation with incomplete and uncertain information: a decision-theoretic approach using a model of the opponent. In: Klusch, M., Kerschberg, L. (Eds.), *Cooperative Information Agents IV: The Future of Information Agents in Cyberspace*, Lecture Notes in Artificial Intelligence, vol. 1860. Springer, Berlin, pp. 107–118.
- Neumann, J.V., Morgenstern, O., 1944. *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, NJ.
- O'Connell, S., O'Sullivan, B., Freuder, E.C., 2002. Strategies for interactive constraint acquisition. In: *Proceedings of CP-02 Workshop on User-Interaction in Constraint Satisfaction*, New York.
- Pawlak, Z., 1982. Rough sets. *Internat. J. Inform. Comput. Sci.* 11 (5), 341–356.
- Pawlak, Z., 1991. *Rough Sets—Theoretical Aspects of Reasonings about Data*. Kluwer, Dordrecht.
- Peasall, J., 2001. *The New Oxford English Dictionary*. Oxford University Press, Oxford.
- Perloff, J.M., 1998. *Microeconomics*. Addison-Wesley, Reading, MA, pp. 29–32 (Chapter 2).
- Pruitt, D., 1981. *Negotiation Behavior*. Academic Press, New York.
- Raiffa, H., 1982. *The Art and Science of Negotiation*. Harvard University Press, Cambridge, USA (sixteenth printing, 2002).
- Restificar, A., Haddawy, P., 2003. Constructing utility models from observed negotiation actions. In: *Proceedings of the Eighteenth Joint International Conference on Artificial Intelligence*, pp. 1404–1405.
- Restificar, A., Haddawy, P., Ha, V., Miyamoto, J., 2002. Eliciting utilities by refining theories of monotonicity and risk. In: *Working Notes of the AAAI'2002 Workshop on Preferences in AI and CP: Symbolic Approach*.
- Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R.D., Shadbolt, N., Velde, W.V.D., Wielinga, B., 2000. *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press, Cambridge, MA.
- Shadbolt, N.R., O'Hara, K., Crow, L., 1999. The experimental evaluation of knowledge acquisition techniques and methods: history, problems and new directions. *Internat. J. Human-Computer Studies* 51 (4), 729–755.

- Steel, P.T., Beasor, T., 1999. *Business Negotiation: A Practical Workbook*. Gower Publishing Limited, Aldershot.
- Turban, E., Aronson, J.E., 1998. *Decision Support Systems and Intelligent Systems*. Prentice Hall, Englewood Cliffs, NJ.
- Unt, I., 1999. *Negotiation Without a Loser*. Copenhagen Business School.
- Volkema, R.J., 1999. *The Negotiation Toolkit: How to get Exactly what you want in any Business or Personal Situation*. Amacom.
- Willmott, S., Calisti, M., Faltings, B., Macho-Gonzales, S., Belakdhar, O., Torrens, M., 2000. CCL: expressions of choice in agent communication. In: *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pp. 325–332.
- Zhu, J., 2004. A buyer-seller game model for selection and negotiation of purchasing bids: extentions and new models. *European J. Oper. Res.* 154 (1), 150–156.