Rewards-Based Negotiation for Providing Context Information

Bing Shi  
State Key Laboratory for Novel Software Technology  
Nanjing University  
Nanjing, China  
shibing@ics.nju.edu.cn

Xianping Tao  
State Key Laboratory for Novel Software Technology  
Nanjing University  
Nanjing, China  
txp@ics.nju.edu.cn

Jian Lu  
State Key Laboratory for Novel Software Technology  
Nanjing University  
Nanjing, China  
lj@nju.edu.cn

ABSTRACT
How to provide appropriate context information is a challenging problem in context-aware computing. Most existing approaches use a centralized selection mechanism to decide which context information is appropriate. In this paper, we propose a novel approach based on negotiation with rewards to solving such problem. Distributed context providers negotiate with each other to decide who can provide context and how they allocate proceeds. In order to support our approach, we have designed a concrete negotiation model with rewards. We also evaluate our approach and show that it indeed can choose an appropriate context provider and allocate the proceeds fairly.

Categories and Subject Descriptors
C.2.4 [Distributed Systems]: Distributed applications — providing context information

General Terms
Context

Keywords
Context-awareness, Context provider, Negotiation

1. INTRODUCTION
Context-awareness is a key concept in pervasive computing. Context informs both recognition and mapping by providing a structured, unified view of the world in which the system operates [1]. Context-aware applications exploit context information, such as location, preferences of users and so on, to adapt their behaviors in response to changing requirements of users and pervasive environments. However, one specific kind of context can often be provided by different context providers (sensors or other data sources of context information) with different quality levels. For example, in a smart home, thermometer A’s measurement precision is 0.1 °C, and thermometer B’s measurement precision is 0.5 °C. Thus A could provide more precise context information about temperature than B. Moreover, sometimes different context providers may provide conflictive context information. For example, different sensors report that the same person is in different places at the same time. Because context-aware applications utilize context information to adapt their behaviors, inappropriate context information may lead to inappropriate behavior. Thus we should design a mechanism to provide appropriate context information for current context-aware applications.

In pervasive environments, context providers considered as relatively independent entities, have their own interests. They hope to get proceeds when they provide context information. However, most existing approaches consider context providers as entities without any personal interests, and use a centralized “arbitrator” provided by the middleware to decide who can provide appropriate context. Thus the burden of the middleware is very heavy, and its decision may be unfair and harm some providers’ interests. Moreover, when such “arbitrator” is broken down, it will cause serious consequences for context-aware applications. In this paper, we let distributed context providers themselves decide who provide context information. Since high reputation could help providers get more opportunities to provide context and get more proceeds in the future, providers try to get the right to provide “good” context to enhance their reputation. In order to get such right, context providers may agree to share some portion of the proceeds with its opponents. Thus context providers negotiate with each other to reach agreement on the issues who can provide context and how they allocate the proceeds. Our approach has some specific advantages:

1. We do not need an “arbitrator” provided by the middleware of pervasive computing to decide who provides context. Thus it will reduce the burden of the middleware.

2. It is more reasonable that distributed context providers decide who provide context, because it can avoid the serious consequences caused by a breakdown of a centralized “arbitrator”.

3. It can guarantee providers’ interests and provide fair proceeds allocation when providers negotiate with each other to reach agreement on their concerned problems.

4. This approach can choose an appropriate provider au-
tomatically. It does not need any applications and users’ intervention.

The negotiation model we have designed to support our approach is also a novel model in negotiation domain. This model can help negotiators reach agreement in the present negotiation process by providing some guarantees over the outcome of next negotiation process (i.e. rewards). Negotiator may find current offer and reward worth more than counter-offer which will delay the agreement, and accepts current offer and reward. Without the reward, it may find current offer worth less than the counter-offer, and proposes its counter-offer. It will cost more time to reach agreement. It also expands the negotiation space considered in present negotiation process, and therefore provides more possibilities to find better agreement.

The remainder of this paper is organized as follows. Section 2 presents some assumptions. Section 3 describes our approach based on negotiation detailedly, including utility functions, negotiation protocol and context providers’ strategies. Section 4 evaluates our approach. In section 5 we introduce some related work and conclude in section 6.

2. SOME ASSUMPTIONS

Before introducing our approach, we would like to give some assumptions:

1. All context providers are well-meaning and honest. During the negotiation process, they exchange information honestly. Rewards confirmed in this negotiation process will be fulfilled in the next negotiation process.

2. All providers must guarantee the system’s interests. They should provide appropriate context information for current applications. After guaranteeing the system’s interest, they can try to maximize their own personal interests. The assumption is reasonable, because when an inappropriate context provider gets the right to provide “bad” context, as a punishment, its reputation will decrease, and the proceeds is also very small.

3. As context providers are independent, factors which influence their negotiation stance and behavior are private and not available to their opponents. Their utility functions are also private.

4. Since the negotiation takes place in pervasive environments, time is a critical factors. The current application often hopes to get context information as quickly as possible, so the time cost to reach agreement should be as short as possible. Context providers often have strict deadline by when the negotiation must be completed.

After presenting these assumptions, we will propose our approach based on negotiation with rewards in the next section.

3. OUR APPROACH

In the beginning, we introduce the concepts of reputation and Quality of Context (QoC) attributes. Both will be used in our approach. Reputation of an agent is a perception regarding its behavior norms, which is held by other agents, based on experiences and observation of its past actions [7]. Here agent means context provider. Each provider’s reputation indicates its historical ability to provide appropriate context information. Quality of Context (QoC) attributes characterize the quality of context information. When applications require context information, they should specify their QoC requirements which express constraints of QoC attributes. Context providers can specify QoC attributes for the context information they deliver. Although we can decide who provides appropriate context according to QoC requirements and context providers’ QoC information, applications’ QoC requirements might not reflect the actual quality requirements. Thus, in addition to QoC, reputation information of context providers is another factor affecting the decision who can provide context information.

Negotiation is a process by which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession making or search for new alternatives [2]. In pervasive environments, all available context providers negotiate with each other to decide who can provide context information. This process will be repeated because a kind of context is needed more than one time. Negotiation using persuasive arguments (such as threats, promises of future rewards, and appeals) allows negotiation parties to influence each others’ preferences to reach better deals effectively and efficiently [9]. This pervasive negotiation is effective in repeated interaction because arguments can be constructed to directly impact future encounters. In this paper, for simplicity, we let negotiation take place between two providers. We extend Raiffa’s basic model for bilateral negotiation [8], and allow negotiators to negotiate with each other by exchanging arguments in the form of promises of future rewards or requests for future rewards. Rewards mean some extra proceeds in the next negotiation process. They can influence outcomes of current and future negotiation.

In our approach, as described by Figure 1, the current application requires Context Manager to provide a specific type of context information satisfying QoC requirements. Context Manager finds that provider A and B can provide such kind of context with different quality levels. Then the manager tells A and B to negotiate to reach agreement on who can provide the context information and how they will allocate the proceeds. Both providers get reputation information from the database Reputation of Context Providers and QoC requirements, and then negotiate with each other according to our negotiation model. When negotiation is completed, the chosen provider will provide the context information to Context Manager, and then Context Manager delivers such information to the application and also stores it in Context Knowledge Base where current and historical context information is stored. The current application gives the feedback information about the provided context, and then Context Manager will update the chosen provider’s reputation information according to the feedback information. Context Manager also provides the proceeds to providers according to the feedback information and the time cost on negotiation. In the following parts of this section, we describe our negotiation model in detail, including context providers’ utility functions to evaluate offers and rewards, negotiation protocol, and strategies to generate offers and rewards.
**3.1 Utility function**

During the negotiation process, one provider proposes an offer and a reward to the other provider. An offer is noted as \( o = (c, p) \): \( c \) indicates the chosen context provider and its domain is \( D_c \) (i.e. the two context providers participating in the negotiation); \( p \) means the proposer’s portion of the proceeds, and its domain is \( D_p = [0,1] \). Its opponent’s portion of the proceeds is \( 1-p \). The reward \( ep \)’s domain is \( D_{ep} = [-1,1] \), and \( |ep| \) means the extra portion of proceeds the proposer promises to provide or requests in the next negotiation process. \( ep < 0 \) means the proposer promises to provide reward, \( ep > 0 \) means the proposer requests reward and \( ep =0 \) means no reward. The opponent evaluates the offer and reward to decide to accept them or propose a counter-offer and a reward. Thus context providers should have utility functions to evaluate offers and rewards.

Time is a critical factor, and only at times in the set \( T = \{0,1,2,…,deadline\} \), context providers can propose their offers. The set \( O \) include all available offers. Context provider \( A \)’s utility function of the offer and reward at time \( t U^A : O \times D_{ep} \times T \rightarrow [-1,1] \) is defined as:

\[
U^A_{o, ep, t} = w^A_1 U^A_{o}(c) + w^A_2 U^A_{p}(p) + w^A_3 U^A_{ep}(ep)) \cdot \delta_A(t)
\]

Similarly, the utility function of \( A \)’s opponent (i.e. \( B \)) can be defined as:

\[
U^B_{o, ep, t} = w^B_1 U^B_{o}(c) + w^B_2 U^B_{p}(p (1-p)) \cdot w^B_3 U^B_{ep}(ep)) \cdot \delta_B(t)
\]

In (1), \( w^A \), \( w^B \) and \( w^A \) are weights given to \( c, p \) and \( ep \) respectively, and \( w^B_1 + w^B_2 + w^B_3 = 1 \). Usually, the context provider pays the most attention to the system’s interests, pays the least attention to the reward, thus \( w^A_1 \geq w^A_2 \geq w^A_3 \). \( U^A_c : D_c \rightarrow [-1,1] \) is the utility function of the issue who provides context. This function is determined by two factors: the distance between \( c \)’s QoC and current application’s QoC requirements, and \( c \)’s reputation. The two negotiators acquire \( c \)’s QoC information from \( c \), and we use the approach proposed in [4] to calculate the distance between \( c \)’s QoC and the application’s QoC requirements. The required context has \( n \) QoC attributes and let the application’s wishes for this context be \( \tilde{d} = (a_1, a_2, … a_n) \) (where \( a_i = \odot \) means the application’s indifference to the \( i \)-th QoC attribute), \( c \)’s QoC attributes \( \tilde{c} = (cp_1, cp_2, … cp_n) \) (where \( cp_e \] = \( \odot \) means \( c \)’s inability to provide a quantitative value for the \( i \)-th QoC attribute). Because numerical distance values of different properties are combined, e.g. location precision in metres with refresh rate in Hz, the standard scale for all dimension is needed. The scaling factors for the QoC attributes are \( \delta = (s_1, s_2, … s_n) \). In addition, different QoC attributes may have different weights: \( w = (w_1, w_2, … w_n) \).

Then \( \tilde{d} = (d_1, d_2, \ldots d_n) \):

\[
d_i = (cp_i - a_i) \cdot s_i \cdot w_i
\]

where \( cp_i - a_i = 0 \) for \( a_i = \odot \) and \( cp_i - a_i = o(a_i) \) for \( cp_i \) = \( \odot \) (\( o(.) \) determines the application’s satisfaction or dissatisfaction when \( c \) is unable to provide an estimate of a QoC attribute, given the value wished for by the application). The distance can be linear distance (1-norm), Euclidean distance (2-norm), or the maximum distance (max-norm):

\[
|\tilde{d}|_1 = |d_1| + |d_2| + \ldots + |d_n| (1-norm)
\]

\[
|\tilde{d}|_2 = \sqrt{|d_1|^2 + |d_2|^2 + \ldots + |d_n|^2} (2-norm)
\]

\[
|\tilde{d}|_\infty = \max\{|d_1|, |d_2|, \ldots |d_n|\} (max - norm)
\]

The detail description of this calculation can be found in [4]. Reputation of \( c \) can be acquired from the database Reputation of Context Providers. \( U^A_{ep} : R \times D_{ep} \rightarrow [-1,1] \) can be defined as:

\[
U^A_{ep}(c) = w^A_1 \cdot U^A_{ep}(d) + w^A_2 \cdot U^A_{ep}(r(ep))
\]

3.2 Negotiation protocol

When provider \( A \) and \( B \) have got QoC requirements and reputation information, they begin to negotiate. They first set their reserved (the lowest acceptable) utility which can guarantee the system’s interests and their personal interests. When the context provider finds the utility of an offer and a reward is lower than its reserved utility, it will reject this proposal and terminate the negotiation process. The provider who starts the negotiation is chosen randomly. We assume \( A \) starts the negotiation, and it proposes offer \( o \) and reward \( ep \) to \( B \) according to its strategy (see subsection 3.3). When \( B \) receives the proposal from \( A \), it uses its utility function to evaluate it. If it is lower than its reserved utility, the provider terminates the negotiation. Otherwise, if

\[
U_B(o, ep, t) \geq U_B(o', ep', t + 1)
\]

i.e. the utility of \( o \) and \( ep \) proposed by \( A \) at time \( t \) is greater than the utility of offer \( o’ \) and reward \( ep’ \) which \( B \) will propose to \( A \) at time \( t + 1 \), \( B \) will accept this offer and reward. The negotiation is completed. However, if

\[
U_B(o, ep, t) < U_B(o', ep', t + 1)
\]
then \( B \) will reject \( A \)'s proposal, and propose its counter-offer and reward to \( A \). When \( A \) receives \( B \)'s counter-offer and reward, \( A \) evaluates them using its utility function, and compares the utility with the utility of offer and reward it wants to propose to \( B \) at time \( t + 2 \), decides to accept it or give its counter-offer and reward. This negotiation process continues and in each negotiation round, context providers concede in order to reach agreement. The negotiation will be successfully finished when agreement is reached, or be terminated forcibly due to deadline or the utility lower than reserved utility. When negotiation is forced to be terminated, Context Manager will ask \( A \) and \( B \) to calculate \( U^A_c(A), U^B_c(B), U^A_d(A) \) and \( U^B_d(B) \) respectively.

\[
U^A_c(A) + U^B_d(A) > U^A_c(B) + U^B_d(B)
\]

Context Manager will provide information blindly in order to enhance its utility. When \( U^A_c(A) + U^B_d(A) < U^A_c(B) + U^B_d(B) \) then \( B \) will get the right to provide context information. When \( U^A_c(A) + U^B_d(A) = U^A_c(B) + U^B_d(B) \) Context Manager will select a provider from \( A \) and \( B \) randomly. In addition, Context Manager allocates the proceeds between the two providers. Although we can select one provider when negotiation is terminated forcibly, however, this may lead to the unfair allocation of the proceeds. Moreover, more time negotiators cost on negotiation, less proceeds will be given. Thus negotiators will try to reach agreement as soon as possible in order to avoid unnecessary loss.

When the negotiation is finished, the chosen provider provides the context information to Context Manager which will deliver the information to current application. According to the application’s feedback information about this context, Context Manager updates the provider’s reputation stored in Reputation of Context Providers. The provider’s reputation may be enhanced or decreased. In addition, according to the feedback and the negotiation time, Context Manager will provide proceeds to the provider. Then the provider will share the proceeds with its opponent according to the negotiation outcome and the reward confirmed in the last negotiation process. For example, in the last negotiation process \( A \) promised to give reward \( ep(0 \leq ep < 1) \) to \( B \), and \( A \)'s portion of the proceeds is \( p \) in current negotiation. Then \( A \)'s actual portion of the proceeds is \( p \cdot (1 - ep) \), and its opponent \( B \)'s portion of the proceeds is \( 1 - p + p \cdot ep \).

### 3.3 Negotiation strategy

The context provider might want to pursue the right to provide context information blindly in order to enhance its reputation. However, when it finally provides “bad” context information, its reputation will be decreased and the proceeds is also very small. Thus the context provider should take action according to its strategy. The aim of provider’s negotiation strategy is to determine the best course of action which will result in a negotiation outcome maximizing its utility function (i.e. how to generate an offer and a reward). In our negotiation model, the context provider generates its offer and reward according to its previous offer and reward and the last one sent by its opponent.

At the beginning of the negotiation, context providers initialize their offers and rewards according to their beliefs and their reserved utility. If context provider \( A \) considers that it can provide “good” context and wants to enhance reputation, then it will propose that \( A \) provides the context information, shares some proceeds with its opponent \( B \), and even promises to give reward. However, if \( A \) considers that it may provide “bad” context, \( A \) will propose that its opponent \( B \) provide the context, and require \( B \) to share some proceeds and provide reward.

During the negotiation process, we assume that at time \( t \) \( A \) proposes offer \( o_t \) and reward \( ep_t \) to \( B \), at time \( t + 1 \), \( B \) proposes counter-offer \( o_{t+1} \) and reward \( ep_{t+1} \) to \( A \). Then at time \( t + 2 \), when the utility of \( B \)'s proposal is greater than \( A \)'s reserved utility, \( A \) gives its response. Now we calculate the expected utility to be conceded at time \( t + 2 \), we use \( Cu \) to express the conceded utility.

\[
Cu = (U_A(o_t, ep_t, t) - U_A(o_{t+1}, ep_{t+1}, t + 1)) \cdot c_A(t + 2)
\]

When \( U_A(o_t, ep_t, t) > U_A(o_{t+1}, ep_{t+1}, t + 1) \), otherwise, \( A \) will accept \( B \)'s proposal where \( c_A : T \rightarrow [0, 1] \) is a monotone-increasing function. \( c_A(t) \) indicates \( A \)'s utility concession rate. \( A \) accepts a little in the beginning before conceding significantly towards the deadline. Then \( A \) generates its offer \( o_{t+2} = (c_{t+2} \cdot p_{t+2}) \) and reward \( ep_{t+2} \) at time \( t + 2 \). The expected utility of \( A \) at time \( t + 2 \) is:

\[
U_A(o_t, ep_t, t + 2) = U_A(o_t, ep_t, t + 2) - Cu
\]

If \( U_A(o_t, ep_t, t + 2) \leq U_A(o_{t+1}, ep_{t+1}, t + 1) \) then \( A \) will accept \( B \)'s proposal (i.e. \( o_{t+1} \) and \( ep_{t+1} \)). Otherwise, \( A \) will propose its counter-offer and reward based on \( Cu \). We assume that \( Cu \) is distributed evenly on \( c \) and \( p \) and \( ep \) (i.e. the utility to be conceded on \( c \), \( p \) and \( ep \) is \( \frac{4}{3}Cu \) respectively). If

\[
|U^A_c(c_t) - U^A_c(c_t) - \frac{4}{3}Cu_{t+2}| \leq |U^A_c(c_{t+1}) - U^A_c(c_t) - \frac{4}{3}Cu_{t+2}|
\]

i.e. the expected utility of \( c \) at time \( t + 2 \) is \( U^A_c(c_t) - \frac{4}{3}Cu_{t+2} \) and it is closer to the utility of \( A \)'s proposal \( c_t \) at time \( t \), then at time \( t + 2 \), \( c_{t+2} = c_t + 1 \), else the utility is closer to \( B \)'s proposal \( c_{t+1} \) and \( c_{t+2} = c_{t+1} \). When \( c_{t+2} \) is equal to \( c_t \), the actual conceded utility of \( c \) is 0, and the total concession of \( p \) and \( ep \) is \( Cu \). We divide the total concession of \( p \) and \( ep \) evenly, and get the conceded utility of \( p \) and \( ep \) respectively.

We calculate \( p_{t+2} \) and \( ep_{t+2} \) as follows:

\[
p_{t+2} = (U^A_p)^{-1}(U^A_p(p_t) - \frac{2}{3}Cu(t + 2))
\]

\[
ep_{t+2} = (U^A_{ep})^{-1}(U^A_{ep}(ep_t) - \frac{2}{3}Cu(t + 2))
\]

When \( c_{t+1} \) is equal to \( c_{t+2} \), the actual conceded utility of \( c \) is \( U^A_c(c_{t+2}) - U^A_c(c_t) \), the total concession of \( p \) and \( ep \) is \( \frac{4}{3}Cu_{t+2} - [U^A_c(c_{t+2}) - U^A_c(c_t)] \), then:

\[
p_{t+2} = (U^A_p)^{-1}(U^A_p(p_t) - \frac{1}{2} \frac{4}{3}Cu(t + 2) - [U^A_c(c_{t+2}) - U^A_c(c_t)])
\]

\[
ep_{t+2} = (U^A_{ep})^{-1}(U^A_{ep}(ep_t) - \frac{1}{2} \frac{4}{3}Cu_{t+2} - [U^A_c(c_{t+2}) - U^A_c(c_t)])
\]

Now, we have generated the offer and reward \( A \) will propose at time \( t + 2 \). Similarly, \( B \) also can generate its offer and reward.

\(^{1}\)For example, \( c_A(t) = \left( \frac{t}{\text{deadline}} \right)^{\frac{1}{\beta}} (0 < \beta < 1) \)
Utility function and weight of $c$, $p$ and $ep$ for each provider

<table>
<thead>
<tr>
<th></th>
<th>$U_c, w_1$</th>
<th>$U_p, w_2$</th>
<th>$U_{rep}, w_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$0.5(1 - \frac{d_A}{500}) + 0.5\frac{rep_A}{1000}$, 0.6</td>
<td>$0.9p$, 0.3</td>
<td>$0.9ep$, 0.1</td>
</tr>
<tr>
<td>B</td>
<td>$0.52(1 - \frac{d_B}{500}) + 0.48\frac{rep_B}{1000}$, 0.5</td>
<td>$0.9p$, 0.45</td>
<td>$0.8ep$, 0.05</td>
</tr>
</tbody>
</table>

Table 1: Utility functions and weights of $c$, $p$ and $ep$ for each provider

4. EVALUATION

In this section, we evaluate the effectiveness of our approach by simulated experiments. Context providers $A$ and $B$ negotiate to reach agreement. They get QoC requirements and calculate the distance between Qoc requirements and their QoC. For simplicity, in our experiments, we assume that the distance has been calculated, and $d_A$ represents distance between QoC requirements and $A$'s QoC, $d_B$ represents distance between QoC requirements and $B$'s QoC. The domain of $d_A$ and $d_B$ is $[0, 500]$. We assume reputation value is a real number and its domain is $[-1000, 1000]$.

rep$_A$ represents $A$'s reputation value and rep$_B$ represents $B$'s reputation value. We assume that both providers pay the most attention to the system's interests, and pay the least attention to the reward, thus $w_2 > w_3 > w_1$, and the weight of $U_a$ approximates the weight of $U_{rep}$. $A$ and $B$'s utility functions and weights of $c$, $p$ and $ep$ are defined in Table 1. We set deadline $t_{deadline} = 100$, and define time discount function $\delta(t)$ and concession rate function $c(t)$ of $A$ and $B$ as follows:

$$\delta_A(t) = 0.9^t$$
$$\delta_B(t) = 0.88^t$$
$$c_A(t) = \left(\frac{t}{t_{deadline}}\right)^{\frac{1}{12}}$$
$$c_B(t) = \left(\frac{t}{t_{deadline}}\right)^{\frac{1}{12}}$$

Given different values of $d_A$, $d_B$, rep$_A$ and rep$_B$, $A$ and $B$ negotiate to reach agreement. The provider that starts the negotiation is chosen at random. We hope that when $d_A < d_B$ and rep$_A \gg$ rep$_B$, $A$ will get the right to provide context and get a major portion of the proceeds, and when $\Delta = d_A - d_B$ is in a small range (e.g. $\pm 50$) and $\Delta_{rep} = rep_A - rep_B$ is in a small range (e.g. $\pm 50$), $A$ and $B$ will get approximately equal opportunities to provide context, and allocate the proceeds evenly. When $\Delta = d_A - d_B$ approximates to $\frac{d_A - d_B}{1000}$ (i.e. the two providers’ abilities to provide context information are approximately equal), we also hope that $A$ and $B$ get equal opportunities to provide context and allocate the proceeds evenly.

According to the three situations above, we make three experiments as follows:

**Experiment 1**: In this experiment, $A$ and $B$ negotiate with each other for 50 times, and at each time, we assign different values to $d_A$, $d_B$, rep$_A$, rep$_B$ (satisfying $d_A < d_B$ and rep$_A \gg$ rep$_B$) and the reserved utilities of $A$ and $B$. When the experiment is completed, we find 3 negotiation games are terminated due to the utility lower than the reserved utility. $A$ gets the right to provide context for 47 times. The average portion of proceeds $A$ get is about 0.683, and $B$'s average portion of proceeds is 0.317. The average time cost to reach agreement is 8.4. We also find that when $B$ asks $A$ to provide context in its first offer, $B$ can require and get more portion of the proceeds because of its goodwill.

**Experiment 2**: $A$ and $B$ also negotiate with each other for 50 times in this experiment given different values of $d_A$, $d_B$, rep$_A$, rep$_B$ (satisfying $-50 \leq \Delta = d_A - d_B \leq 50$ and $-50 \leq \Delta_{rep} = rep_A - rep_B \leq 50$) and the reserved utilities of $A$ and $B$. After the experiment, we find that there are 8 negotiation games terminated due to the utility lower than the reserved utility. $A$ and $B$ get the right to provide context for 20 times and 22 times respectively. The average portion of proceeds $A$ get is 0.528 and $B$'s average portion of the proceeds is 0.472. The average time cost on negotiation is 10.5.

**Experiment 3**: In this experiment, $A$ and $B$ also negotiate with each other for 50 times given $d_A$, $d_B$, rep$_A$, rep$_B$ (satisfying $-0.2 \leq \frac{d_A - d_B}{1000} \leq \frac{d_A - d_B}{1000} \leq 0.2$) and the reserved utilities of $A$ and $B$. There are 6 negotiation games terminated forcibly. $A$ and $B$ get the right to provide context for 21 times and 23 times respectively. The average portion of proceeds $A$ get is 0.481 and $B$'s average portion of the proceeds is 0.519. The average time cost on negotiation is 9.2.

One thing should be mentioned is that except for $d$, rep, $p$ and $ep$, other factors (e.g. weights, time discount function $\delta(t)$ and concession rate function $c(t)$) could also affect the negotiation outcome. These factors should be adjusted according to providers’ beliefs at the beginning of each negotiation process. In our experiments, for similarity, we assign values to them without any particularity in advance. These experiments’ results prove that our approach can choose an appropriate context provider and can provide a relatively fair proceeds allocation. When one provider is obviously more appropriate than the other provider, the provider will get the right to provide context and get a major portion of the proceeds. When both providers have the approximately same abilities to provide context, their opportunities to provide context are equal and they can get about a half portion of the proceeds respectively.

5. RELATED WORK

In [4], Huebischer and McCann have proposed an adaptive middleware design for context-aware applications. Their adaptive middleware uses utility functions to choose the best context provider (given the QoC requirements of applications and the QoC of alternative means of context acquisition).

In our negotiation model, the calculation of utility function $U_c$ was inspired by this approach. Henriksen and Indulska propose an approach to modelling and using imperfect information in [3]. They characterize various types and sources of imperfect context information and present a set of novel context modelling constructs. They also outline a software infrastructure that supports the management and use of imperfect context information. Judd and Steenkiste in [5] describe a generic interface to query context services allowing clients to specify their quality requirements as bounds on accuracy, confidence, update time and sample interval. In [6], Lei et al. present a context service which accepts fresh and confident meta-data from context sources, and passes this along to clients so that they can adjust their level of trust accordingly. [10] presents a framework for realizing dynamic context consistency management. The framework supports inconsistency detection based on a semantic
matching and inconsistency triggering model, and inconsistency resolution with proactive actions to context sources.

Most approaches to provide appropriate context utilize a centralized “arbiter”. In our approach, we let distributed context providers themselves decide who can provide appropriate context information. Our approach can reduce the burden of the middleware, because we do not need the middleware to provide a context selection mechanism. It can avoid the serious consequences caused by a breakdown of the “arbiter”. Also, it can guarantee context providers’ interests.

6. CONCLUSION AND FUTURE WORK

How to provide the appropriate context information is a challenging problem in pervasive computing. In this paper, we have presented a novel approach based on negotiation with rewards to attempt to solve such problem. Distributed context providers negotiate with each other to reach agreement on the issues who can provide the appropriate context and how they allocate the proceeds. The results of our experiments have showed that our approach can choose an appropriate context provider, and also can guarantee providers’ interests by a relatively fair proceeds allocation.

In this paper, we only consider how to choose an appropriate context provider from two providers. In the future work, this negotiation model will be extended, and more than two context providers can negotiate with each other to decide who is the most appropriate context provider. In the extended negotiation model, how to design efficient negotiation strategies will be a challenging problem. We assume that the context provider will fulfill its promise of reward in the next negotiation process. In fact, the context provider might deceive its opponent and provide illusive promise. We should solve this problem in the future. We also should deal with interactions which are interrupted by failing communication links in the future work.

7. ACKNOWLEDGEMENT

The work is funded by 973 Project of China(2002CB312002, 2006CB303000), NSFC(60403014) and NSFJ(BK2006712).

8. REFERENCES