A Utility based, Multi-Attribute Negotiation Approach for Semantic Web Services

SANDEEP KUMAR
WSEAS Research Department
Agiou Ioannou Theologou 17-23
Zografou, 15773, Athens, Greece
also with the
Department of Information Technology,
M. M. Engineering, M. M. University,
Mullana, Ambala, India
sandeepkumargarg@gmail.com

NIKOS E. MASTORAKIS
WSEAS Research Department
Agiou Ioannou Theologou 17-23
Zografou, 15773, Athens, Greece
also with the
Technical University of Sofia,
Industrial Engineering Department
Sofia 1000, Bulgaria
mastor@wseas.org

Abstract: Apart from some other important Semantic Web service related processes such as discovery, selection, composition etc., the process of negotiation is also generally required in the semantic web based systems. Before taking the services of a service provider, the service requester may need to negotiate with it on various issues. A utility-based, multi-attribute negotiation approach capable of providing negotiation between participating semantic web services has been presented in this paper. The approach is based upon the use of utility functions in the negotiation process and uses multiple attributes as the basis of negotiation. A communication model describing the negotiation process has been presented. The paper also presents the algorithms for various activities involved in the negotiation process. The work also proposes a novel concept of negotiation-feedback using a novel data-structure, Agreement Table. This concept can be helpful in expediting the negotiation process by decreasing the number of negotiation steps in which the agreement is reached. An evaluation of the work has been presented and a prototype system providing negotiation between semantic web services has been implemented.

Key-Words: - semantic web; utility; semantic web service, negotiation.

1 Introduction

Before taking the services from service provider (SP), in addition to performing the discovery, selection and composition processes, the service requester (SR) may also needs to perform the negotiation with the SP to establish an agreement over the various service-attributes such as price, quality, time-period, reliability etc. Negotiation is the process by which two or more parties make joint decision. The involved parties first verbalize demands and then move toward an agreement through a process of concession formation or search for new alternatives [1]. A lot of works related to the negotiation process have been reported in the literature such as ([2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], and [16]). But, most of them either not considers the negotiation from the perspective of SWSs or only deals with the theoretical
aspects of negotiation between SWSs. This paper mainly focuses on the presentation of an approach for utility-based negotiation. The paper also provides an introduction to a computational model for utility-calculation ([17], [18]). This work will provide the negotiation approach based upon the utility model presented by these earlier works.

The main contribution of paper is listed as below:
- A utility based negotiation approach for negotiation between SWSs.
- The algorithms for various activities involved in the negotiation process along with the communication model for negotiating services have been presented.
- A novel concept of negotiation-feedback using a novel data-structure, Agreement-Table, has been proposed which can expedite the negotiation process.
- The work has been evaluated and a system providing negotiation between semantic web services (SWSs) has been implemented.

The remainder of this paper is organized as follows. Section-2 presents an introduction to some of similar existing works. The proposed utility based negotiation approach has been presented in the Section-3. Section-4 introduces a utility-computation model. The work has been evaluated and a negotiation based system has been implemented in the Section-5. The work has been concluded in the Section-6.

2. Related Works

From the last decade, the automated negotiation in the multi-agent systems and the web services has obtained the attention of large community of researchers. Especially, a lot of reported works are available on the negotiation in multi-agent systems. Among others, some of the works to name are ([2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], and [13]). Out of these, few of the works present multi-attribute negotiation based on the utility theory. The works, ([2], [3], [4], [5], [8], [11], and [12]), present the approaches for multi-attribute negotiation based on utility theory. But, lesser works are available on the negotiation in the perspective of SWSs. Some of the works in this field are ([14], [15], and [16]).

[2] in their works have presented a multi-attribute negotiation approach and a utility model for multi-agent negotiation. They have designed the utility model especially suited for the telecommunication services. The work in [3] presents a component-based generic agent-architecture for multi-attribute negotiation. They have used the concept of utility theory and have used the combination of ease utility and financial utility in the negotiation process. Their approach provides cooperative negotiation between the participating agents and the agents are able to use any amount of incomplete preference information revealed by the negotiation partner. [4] have presented a multi-dimensional, multi-step negotiation mechanism for task allocation among the cooperative agents. Their mechanism is based on the distributed search. It uses the concept of marginal utility gain and marginal utility cost to structure the search process and to find the solution which maximizes the agents’ combined utility. The work uses multi-attribute utility functions into the negotiation process. SCENS, a Secure Content Exchange Negotiation System, enables the sharing of sensitive multimodal digital data residing in the distributed digital repositories [5]. The work proposes the use of utility functions in the negotiation process. [8] in their works have presented an agent based, multi-attribute negotiation model for large-scale construction project supply chain coordination. They have used the concept of utility theory. Their negotiation models consist of three processes: attributes evaluation, utility determination, and attribute planning. [11] have presented a multi-attribute negotiation model focused on the situation when negotiation parties have incomplete information. In their model, a nonbiased mediator has been adopted who applied the query learning to maintain near Pareto-efficiency. This model can be applied to the situations where an agent’s preference can be neither explicitly characterized by a utility function, nor known by each other. In this model, the n-dimensional negotiation space has been decomposed into the negotiation base lines to reduce the negotiation complexity. [12] have extended the work in [11] to propose a framework for automated multi-attribute negotiation. They have dealt with the issues of incomplete information, Pareto optimality, and tractability. In their framework, the negotiation strategy of an agent is composed of three components:
conceding strategy, responding strategy, and proposing strategy. In the proposing strategy, they have presented two proposing mechanisms: shortest-distance proposing mechanism and Pareto optimal mediating mechanism. The shortest-distance proposing mechanism is applied when the agents know their own utility function but do not know the utility function of opponent agent, while the Pareto optimal mediating mechanism is used when the agents know neither of utility functions. In our proposed approach also, the agents know their own utility functions only but do not know the utility functions of opponent agents. [14] have presented an approach for negotiation among SWSs in the form of negotiation protocols and strategies by adapting and refining Rubinstein’s alternating offers protocol [19] to web service negotiation. They have provided the adaptation of alternating offers negotiation language, alternating offers negotiation protocol, and alternating offers negotiation strategies to the web services domain. The work by [15] presents the use of Peertrust language for deciding if the trust can be established between the service requester and the service provider. They have discussed the use of different matchmakers together with service registries in order to allow users to find the services fulfilling their goals. They have added the modeling elements to the WSMO (Web Service Modeling Ontology) [20] for including the trust information in the description of SWSs. The work in [16] presents the use of PeerTrust for trust negotiation between peers on the semantic web and to control the use of different access-control policies. They have presented the use of PeerTrust to support delegation, policy protection, and negotiation strategies. A PeerTrust automated trust negotiation engine has also been presented. The work also presents the syntax and semantics of GDLP (Guarded Distributed Logic Programs) and shows the way of representing the appropriate policies and negotiation rules for automated trust negotiation using it. Table 1 shows the summary of various related works discussed above.

In this paper, we will present an approach for multi-attribute negotiation between SWSs based on the utility theory. In contrast to the other related works on negotiation between SWSs such as ([14], [15], [16]), which only discusses the theoretical aspects, our work presents the detailed computation model for calculation of utility of SP and SR. The work also presents the algorithms for various activities involved in the proposed approach and a communication model for this using FIPA Communicative Acts [21]. The computation models for some of important parameters of negotiation process such as opportunity-cost, opportunity-gain, and negotiation-effort have also been proposed.

3 Utility-based Negotiation Approach

In this section, we have proposed a utility based approach for negotiation between SWSs. It involves the process of offering proposals with incremental concession from both SR and SP to each other until an acceptable agreement is obtained or the numbers of negotiation steps exceed the threshold limit. The acceptability of proposal is checked based upon the utility of SR and SP. A computational model for calculation of utilities of SR and SP ([17], [18]) is introduced in the next section. A communication model for negotiation between SP and SR has also been presented.

3.1 Communication Model

The proposed negotiation approach involves the use of multiple attributes of SWSs for negotiation. The proposal between SP and SR contains the values for multiple attributes and the decision of agreement is taken based upon their combined value. A utility value is used which is dependent on the values of all the attributes and represents the preference of corresponding SWS. Utility theory is the appealing form of representing inputs to decision-making under uncertainty for automated systems because it can readily be mapped onto numerical optimization-based approaches [22]. The initial values of various attributes and conditions for termination of negotiation between SWSs can be fetched from their corresponding service profiles. The communication model for the proposed utility model is shown in Figure 1. Figure shows the communication between SR and SP during the negotiation using Communicative Acts of FIPA [21]. As shown in Figure 1, the negotiation process starts with the request from SR to SP for providing the services. If the request is refused by the SP, the process is terminated. But, if the SP agreed to provide services, the SR sends a call to SP to send an initial proposal for starting the negotiation. At this step also, if the call for initial proposal is refused by the SP, then negotiation process got terminated, otherwise SP responses with an
initial proposal to the SR. Now, if this proposal is acceptable to the SR, then it is informed to the SP. SP informs the SR about various parameters of agreement and the negotiation is terminated. In the case of rejection by SR, the SR sends a new proposal to SP. Now, SP checks the proposal and if acceptable, informs the SR with acceptance. The values of various agreement-parameters are informed by the SR to SP and the process is terminated. But in the case of rejection by SP, a new proposal is sent by the SP to SR. This process continues until either the proposal acceptable to both SP and SR occurs or the number of negotiation-steps exceeds the threshold limit. In the presented negotiation approach, the utility values for SR and SP can be calculated using the utility calculation models presented in [17] and [18].
3.2 Negotiation Environment

Figure 2 shows the environment in which the proposed utility based negotiation is performed between SWSs.

The environment contains a set of SPs that offer computer-based services to their clients i.e. SRs, which may themselves be service providers. Each SP is an independent entity with attached service profiles and motivated by some business concerns such as
achieving profitability and hence demands some payment for providing services. However, to keep the things simple, only a single SP is shown in the Figure 2.

3.3 Agreement Table

The proposed negotiation approach also involves a feedback-system, which on successful negotiation stores the agreement into the Agreement-Table (AT). AT is a data-structure maintained by the SP in its service profile and holds the values of various attributes of the latest agreement with a SR. An example AT is shown in the Figure 3. Each entry of AT for a SP contains following elements:

i. Service Requester Identifier (SR)
ii. Agreement values for the latest agreement between the corresponding SR and given SP.

The values stored in the AT can be used in the future negotiations. For example, in the case a SR, which has taken the services from the reference SP in past, request SP for negotiation to take its services, then SP can fetch the already stored agreement from the AT corresponding to given SR and can start negation from this agreement. This will have high possibility that this agreement will be acceptable to SR in first offer or it will be acceptable in a few negotiation steps. Thus, a lot of time and efforts will be saved.

3.4 Various Algorithms

The algorithms for generating a new proposal by SP and SR are shown in the Figure 4 and Figure 5 respectively. Figure 6 shows the algorithm for checking the acceptance of offer of SP/SR by SR/SP. The algorithm uses a function for calculating the utility of SR/SP, the detailed implementation of which will be described in the next sub-section. It is to mention that the method for calculation of utility is different for SP and SR. The algorithm for checking the termination conditions of the negotiation process is shown in Figure 7. The negotiation process is terminated when either the acceptable offer is obtained or the number of negotiation steps exceeds a threshold. As algorithm shows, the number of negotiation steps is decided by the values of the variables which are used to increase or decrease the initial attribute-values. Smaller the values of these variables, more will be the number of steps permissible in negotiation process.
**Algorithm: Generation of New Proposal by SP**

| Standard proposal: $stan_p$ (standard price), $stan_q$ (standard quality), $stan_t$ (standard time-period) |
| Current proposal: $pro_p$ (proposed price), $pro_q$ (proposed quality), $pro_t$ (proposed time-period) |
| Previous proposal: $pre_p$ (previous price), $pre_q$ (previous quality), $pre_t$ (previous time-period) |
| $delt_p$: a short price-value |
| $delt_q$: a short quality-value |
| $delt_t$: a short period of time |
| $ratio_p$: a small number used to increase the standard price |
| $ratio_q$: a small number used to decrease the standard quality |
| $ratio_t$: a small number used to increase the standard time-period |

begin

if (first proposal)

//set values for the first proposal from SP

$pro_p = ratio_p \times stan_p$;

$pro_q = ratio_q \times stan_q$;

$pro_t = ratio_t \times stan_t$;

else

//set values for other new proposals from SP in due course of negotiation

if ($pre_p > stan_p$)

$pro_p = pre_p - delt_p$;

$pro_q = pre_q$;

$pro_t = pre_t$;

else

if ($pre_t > stan_t$)

$pro_p = pre_p$;

$pro_q = pre_q$;

$pro_t = pre_t - delt_t$;

else

if ($pre_q < stan_q$)

$pro_p = pre_p$;

$pro_q = pre_q + delt_q$;

$pro_t = pre_t$;

end-if

end-if

end


**Algorithm: Generation of New Proposal by SR**

| Standard proposal: $stan_p$ (standard price), $stan_q$ (standard quality), $stan_t$ (standard time-period) |
| Current proposal: $pro_p$ (proposed price), $pro_q$ (proposed quality), $pro_t$ (proposed time-period) |
| Previous proposal: $pre_p$ (previous price), $pre_q$ (previous quality), $pre_t$ (previous time-period) |
| $delt_p$: a short price-value |
| $delt_q$: a short quality-value |
| $delt_t$: a short period of time |
| $ratio_p$: a small number used to decrease the standard price |
| $ratio_q$: a small number used to increase the standard quality |
| $ratio_t$: a small number used to decrease the standard time-period |

begin

if (first proposal)

//set values for the first proposal from SR

$pro_p = ratio_p \times stan_p$;

$pro_q = ratio_q \times stan_q$;

$pro_t = ratio_t \times stan_t$;

else


// Set values for other new proposals from SR in due course of negotiation
if (pre\textsubscript{p} < stan\textsubscript{p})
    pro\textsubscript{p} = pre\textsubscript{p} + delt\textsubscript{p};
    pro\textsubscript{q} = pre\textsubscript{q};
    pro\textsubscript{t} = pre\textsubscript{t};
else
    if (pre\textsubscript{t} < stan\textsubscript{t})
        pro\textsubscript{p} = pre\textsubscript{p};
        pro\textsubscript{q} = pre\textsubscript{q};
        pro\textsubscript{t} = pre\textsubscript{t} + delt\textsubscript{t};
    else
        if (pre\textsubscript{q} > stan\textsubscript{q})
            pro\textsubscript{p} = pre\textsubscript{p};
            pro\textsubscript{q} = pre\textsubscript{q} - delt\textsubscript{q};
            pro\textsubscript{t} = pre\textsubscript{t};
        end-if
    end-if
end-if
end

Figure 5: Generation of New Proposal by SR

Algorithm: Checking Proposal
received proposal: rec\textsubscript{p} (price in received proposal), rec\textsubscript{q} (quality in received proposal), rec\textsubscript{t} (time-period in received proposal)
utility\textsubscript{v}: variable to store utility value
begin
    utility\textsubscript{v} = calculate\_utility(rec\textsubscript{p}, rec\textsubscript{q}, rec\textsubscript{t});
    // Detail procedure for calculate\_utility() function is described in next sub-section.
    // The formulation for calculate\_utility() is different for SP and SR
    if (utility\textsubscript{v} >= 1)
        received proposal is acceptable;
    else
        received proposal is not acceptable;
    end-if
end

Figure 6: Checking the proposal for acceptance

Algorithm: Checking Termination Condition for Negotiation Process
utility\textsubscript{v}: utility value for the received proposal
standard proposal: stan\textsubscript{p} (standard price), stan\textsubscript{q} (standard quality), stan\textsubscript{t} (standard time-period)
latest sent proposal: last\textsubscript{p} (price in latest sent proposal), last\textsubscript{q} (quality in latest sent proposal), last\textsubscript{t} (time-period in latest sent proposal)

begin
    if (utility\textsubscript{v} >= 1)
        // utility more than or equal to 1 implies that the received proposal is acceptable
        // so accept the proposal and terminate negotiation process with agreement
        terminate negotiation
    end-if
    if (last\textsubscript{p} = stan\textsubscript{p} AND last\textsubscript{q} = stan\textsubscript{q} AND last\textsubscript{t} = stan\textsubscript{t})
        // negotiation-steps exceed the maximum threshold limit.
        // the number of steps in threshold limit is decided by the values of
        terminate negotiation
end

### 4 Computational Model for Calculation of Utilities

In this section, a computational model for the calculation of utilities ([17], [18]) of both SR and SP has been presented. The section also presents the formalization of various parameters using which the introduced model can be extended.

The utility of a SR/SP represents its happiness or preference. The utility function should be designed in such a way that it produces such numerical value which increases or decreases to represent the more or less happiness or preference of SR/SP. The SR/SP should be indifferent to (i.e. equally happy with) the various combinations of values of the different attributes in proposal which produces same utility [22]. So, utility function should be such that it produces same utility-value for this type of combinations. For a proposal with attributes (price, quality), if the quality is improved then the corresponding price can also be increased in the appropriate ratio. Where, quality is manifested as the service usability and utility and includes various factors such as performance, integrity, accessibility, availability, interoperability, security etc. ([23], [24]). It represents the totality of features and characteristics of a service. So, if the price has been increased in the required ratio only, then the utility should remain the same. Let us take a simple example. Let, a SWS has utility \( u \) for a proposal \( pr \) with price-value \( p \) and quality \( q \). Further, price and quality are related to each other by one-to-one ratio i.e. the ratio in which quality is increased, the price also get increased by same ratio. So, if quality is improved to \( 1.5q \) and the price also increased to \( 1.5p \), then the utility-value should remain the same i.e. \( u \). Our proposed utility function is based on the same understanding.

The utility of a SWS depends upon the values of different attributes of the service. The presented utility function is dependent on the multiple attributes of the service. We have considered three main attributes of a service i.e. price, quality, time-period (response-time). So, utility can be expressed as a function of these attributes, \( utility(price, quality, response\text{-}time) \). Further, the price of a service depends upon the quality of service and the response-time. It is likely that the service provider will expend more resources to provide a higher quality or to complete request in lesser response-time, and to maintain profitability it will want to recoup its extra cost by raising the price of the service [22]. The service provider may also require more response-time if the quality-level is increased. So, following relations will hold:

\[
price = f(quality) \quad \text{Or} \quad price \propto quality \quad \ldots (1)
\]

\[
price = f \left( \frac{1}{response\text{-}time} \right) \quad \text{Or} \quad price \propto \frac{1}{response\text{-}time} \quad \ldots (2)
\]

\[
response\text{-}time = f(quality) \quad \text{Or} \quad response\text{-}time \propto quality \quad \ldots (3)
\]

If \( P_{\text{initial}} \), \( Q_{\text{initial}} \), \( T_{\text{initial}} \) be the price, quality, and response-time of a service, then using the above discussion, the relations for calculating the new price and new response-time of a service on changing the quality can be derived as below.

If \( Q_{\text{new}} \) is the new quality required, then the percentage-change in quality can be represented as follows:

\[
\Delta Q = \frac{Q_{\text{new}} - Q_{\text{initial}}}{Q_{\text{initial}}} \times 100 \quad \ldots (4)
\]

Where, \( \Delta Q \) is the percentage-change in quality and holds \( 0 \leq \Delta Q \leq 100 \).

Now, using relation (3), the percentage-change in response-time can be calculated as:
\[
\Delta T = \frac{K_{TQ} \cdot \Delta Q}{100} \quad \ldots (5)
\]

Where, \( \Delta T \) is the percentage-change in time due to quality-change and it holds \( 0 \leq \Delta T \leq 100 \). \( K_{TQ} \) is a constant, which satisfies the relation, \( 0 \leq K_{TQ} \leq 100 \). Its value is decided by the service provider. It represents the percentage of the percentage-change in quality (\( \Delta Q \)) with which the response-time should be changed. It means that if the percentage-change in quality (\( \Delta Q \)) is 80 and \( K_{TQ} \) is equal to 70, then the percentage-change in price (\( \Delta Pq \)) will be 56. The \( K_{TQ} = 100 \) implies that the price and quality has one-to-one ratio and price should be equally changed as change in quality. The \( K_{TQ} = 0 \) implies that price is not dependent on the quality.

So, the new response-time will be:

\[
T_{new} = T_{initial} + \left( T_{initial} \cdot \frac{\Delta T}{100} \right) \quad \ldots (6)
\]

In other form, the \( T_{new} \) can be represented as:

\[
T_{new} = T_{initial} + \left( T_{initial} \cdot \frac{K_{TQ}}{100} \left( \frac{Q_{new} - Q_{initial}}{Q_{initial}} \right) \cdot 100 \right) \quad \ldots (7)
\]

Now, based on the relation (1), the percentage-change in price due to quality-change can be derived as follows:

\[
\Delta Pq = \frac{K_{PQ} \cdot \Delta Q}{100} \quad \ldots (8)
\]

Where, \( \Delta Pq \) is the percentage-change in price due to quality-change and it holds \( 0 \leq \Delta Pq \leq 100 \). \( K_{PQ} \) is a constant, which satisfies the relation, \( 0 \leq K_{PQ} \leq 100 \). Its value is decided by the service provider. It represents the percentage of the percentage-change in quality (\( \Delta Q \)) with which the price should be changed. It means that if the percentage change in quality (\( \Delta Q \)) is 80 and \( K_{PQ} \) is equal to 70, then the percentage-change in price (\( \Delta Pq \)) will be 56. The \( K_{PQ} = 100 \) implies that the price and quality has one-to-one ratio and price should be equally changed as change in quality. The \( K_{PQ} = 0 \) implies that price is not dependent on the quality.

So the new price, after taking the quality-change into consideration, will be:

\[
P_{new} = P_{initial} + \left( P_{initial} \cdot \frac{\Delta Pq}{100} \right) \quad \ldots (9)
\]

In other form, the \( P_{new} \) can be represented as:

\[
P_{new} = P_{initial} + \left( P_{initial} \cdot \frac{K_{PQ}}{100} \left( \frac{Q_{new} - Q_{initial}}{Q_{initial}} \right) \cdot 100 \right) \quad \ldots (10)
\]

The equations (9) and (10) represent the new price after considering the effect of quality-change. On changing the quality, if the response-time has been changed according to the equation (7), then there should not be any change in price due to response-time change, but if the change in response-time is not according to the equation (7), then this change of response-time from the \( T_{new} \) will also affect the price.

The percentage-change in price due to change in response-time can be calculated as follows:

If actual new response-time is \( T_{ANew} \), then the percentage-change in response-time from the required response-time \( T_{new} \) will be:

\[
\Delta Ta = \frac{T_{new} - T_{ANew}}{T_{new}} \cdot 100 \quad \ldots (11)
\]

Where, \( \Delta Ta \) is the percentage-change of the response-time from the required response-time \( T_{new} \) and it holds \( 0 \leq \Delta Ta \leq 100 \).
Now, based on the relation (2), the percentage-change in price due to change in the required response-time can be calculated as follows:

$$\Delta P_t = \frac{K_{PT} \cdot \Delta T_a}{100} \quad \ldots (12)$$

Where, $\Delta P_t$ is the percentage-change in price due to change in response-time and it holds $0 \leq \Delta P_t \leq 100$. $K_{PT}$ is a constant, which satisfies the relation, $0 \leq K_{PT} \leq 100$. Its value is decided by the service provider. It represents the percentage of the percentage-change in response-time ($\Delta T_a$) with which the price should be changed. It means that if the percentage-change in response-time ($\Delta T_a$) is 30 and $K_{PT}$ is equal to 30, then the percentage-change in price ($\Delta P_t$) will be 9. The $K_{PT} = 100$ implies that the price and response-time has one-to-one ratio and price should be equally changed as change in response-time. The $K_{PT} = 0$ implies that price is not dependent on the change in response-time.

So, the new price after taking the effect of change in required response-time will be:

$$P_{new} = P_{initial} + \left( \frac{P_{initial} \cdot \Delta P_t}{100} \right) \quad \ldots (13)$$

In other form, the $P_{new}$ can be represented as:

$$P_{new} = P_{initial} + \left( \frac{P_{initial} \cdot K_{PT} \left( \frac{T_{new} - T_{Anew}}{T_{new}} \right) \ast 100}{100} \right) \quad \ldots (14)$$

It can be inferred from equations (13) and (14) that if the actual response-time ($T_{Anew}$) is more than the required response-time ($T_{new}$), then the price should be decreased, but if the actual response-time ($T_{Anew}$) is less than the required response-time ($T_{new}$), then the price should be increased.

The new price, after considering the effect of change in quality as well as the effect of change in response-time, will be:

$$P_{new} = P_{initial} + \left( \frac{K_{PT} \left( \frac{Q_{new} \ast Q_{initial}}{100} \right) + \left( \frac{K_{PT} \left( \frac{T_{new} - T_{Anew}}{T_{new}} \right) \ast 100}{100} \right)}{100} \right) \quad \ldots (15)$$

The above derived equations can be used for the calculation of utility for SR and SP. Consider that SR has some proposal ($P_{initial}$, $Q_{initial}$, $T_{initial}$) and values of various constants $K_{PT}$, $K_{PQ}$, $K_{TQ}$ on which SR agrees. These values can be maintained in the service profile of SR. Let, ($P_{offer}$, $Q_{offer}$, $T_{offer}$) be the proposal obtained by SR from SP. The offered quality, $Q_{offer}$, can be treated as the new quality, $Q_{new}$, and offered response-time, $T_{offer}$, as the actual response-time, $T_{Anew}$. Then using equations (4) to (15), the value for required price $P_{new}$ can be calculated, which is the value of price considered appropriate by the SR for given quality and given response-time. This value of price, $P_{new}$, which has been calculated by considering the effect of both quality-change and change in response-time, will represent the preferred-level of SR or the level at which SR is happy for given quality and response-time. Whereas, $P_{offer}$ is the offered price for given quality and response-time. So, the ratio of $P_{new}$ and $P_{offer}$ will represent the happiness/preference level of SR. On the other hand, the utility of SR/SP for a service also represents their respective happiness/preference (Wilkes, 2008). Hence, the utility of SR can be represented as:

$$Utility_{SR} = \frac{P_{new}}{P_{offer}} \quad \ldots (16)$$

From the equation (16), it can be inferred that if the offered-price is more than the required price, then the utility of SR will be less than one and the proposal will not be accepted. In the similar fashion, the utility for SP can be calculated. The only difference is that in the
case of SP, for the proposal to be acceptable the offered price should be more than or equal to the required price. Hence, the utility of SP can be represented as:

$$ Utility_{SP} = \frac{P_{offer}}{P_{new}} \quad \ldots (17) $$

From the equation (17), it can be inferred that if the offered price is less than the required price, then the utility of SP will be less than one and the proposal will not be accepted.

### 4.1 A Possible Extension

The negotiation process and the utility model presented above can be extended by considering some other important parameters such as opportunity-cost, opportunity-gain, and negotiation-cost [4]. During the negotiation process, when the SP makes a commitment to perform a task, it looses the opportunity to perform another incoming task with higher utility. This loss occurring to the SP on committing a negotiation can be called as Opportunity-Cost. So, in the utility-calculation for the SP, not only the actual usage of resources should be considered, but the opportunity-cost should also be involved. Hence, the net utility of SP at a negotiation step should be calculated by deducting the opportunity-cost from the utility-value [4]. The opportunity-cost of SP at a negotiation-step will depends upon the utility gained by the SP at that step. So, the opportunity-cost can be represented as below:

$$ Opportunity – Cost = \frac{K_{OC} \cdot Utility_{SP}}{100} \quad \ldots (18) $$

Where, $K_{OC}$, with value ranging between $0 \leq K_{OC} \leq 100$, is the constant representing the percentage value.

On the other hand, on importing the task to SP, SR leaves itself more freedom to accept another task of higher utility. This gain occurring to the SR on importing the task to SP can be called as Opportunity-Gain. So, in the utility-calculation for the SR, not only the gain from getting the task done from SP should be considered, but the opportunity-gain should also be involved. Hence, the net utility of SR at a negotiation step should be calculated by adding the opportunity-gain with the utility-value [4]. The opportunity-gain of SR at a negotiation-step will depends upon the utility gained by the SR at that step. So, the opportunity-gain can be represented as below:

$$ Opportunity – Gain = \frac{K_{OG} \cdot Utility_{SR}}{100} \quad \ldots (19) $$

Where, $K_{OG}$, with value ranging between $0 \leq K_{OG} \leq 100$, is the constant representing the percentage value.

The negotiation process consumes resources such as time, computational capability, communication capacity etc. These resources otherwise could be used for some other tasks. Also, in some cases, the negotiation process has an influence over the process and time of execution of task. This can also reduce the utility of process. These losses caused by the negotiation process can be termed as Negotiation-Cost or Negotiation-Effort. Negotiation-Effort can be measured by the number of negotiation steps and it increases with increase in the number of negotiation steps. To make the negotiation-effort and utility gain comparable, the number of negotiation steps can be mapped into a certain percentage of initial utility without negotiation [4]. So, the negotiation-effort can be represented by following relation:

$$ Negotiation – Effort = \frac{K_{NC} \cdot N \cdot Utility_{initial}}{100} \quad \ldots (20) $$

Where, N has the value for number of negotiation-steps. $Utility_{initial}$ is the initial utility of SR/SP if the task has been performed without negotiation. $K_{NC}$ is the percentage-value which can be chosen by SR/SP. It decides how the negotiation-effort would affect the utility of SR/SP. The equation (20) implies that the each step of negotiation decreases the achieved utility by a value equal to the $K_{NC}$ percentage of $Utility_{initial}$.

Hence, on considering the opportunity-cost, opportunity-gain, and negotiation-effort, the utilities for SR and SP can be given as below:
Net Utility for SP = Utility_{SP} - Negotiation-Effort - Opportunity-Cost
Net Utility for SR = Utility_{SR} - Negotiation-Effort + Opportunity-Gain

5 Evaluation and Implementation

The work mainly presents a utility based negotiation approach for SWSs. The proposed approach can be evaluated by comparing it with existing similar works. The proposed MAN mainly focuses on the presentation of communication model and utility model for negotiation process. The paper presents a utility based multi-attribute negotiation for SWSs. Many reported works are available on the utility based multi-attribute negotiation for multi-agent systems, but a little works are only available providing negotiation strategies between SWSs. Remainder of this section presents the evaluation of proposed work by comparing it with existing similar works.

[3] have presented the utility based multi-attribute negotiation for multi-agent systems. They have presented the concept of financial utility and ease utility in the negotiation process. But, their work does not consider the negotiation from the perspective of SWSs. Also, they have not used the concept of storing the successful agreements for future use. Similarly, the work by [4] has presented the multi-dimensional, multi-step, multi-attribute negotiation from multi-agent perspectives only. Their work also suffers from the same drawback as that of work by [3]. [5] in their work have presented a Secure Content Exchange Negotiation System (SCENS) for multi-agent systems which consists of the three layers: layer one for web-based negotiation support system, layer two providing negotiation web services to end user, and layer three providing open and automated negotiation environment. They have discussed only first two layers, but have not provided details on the negotiation and communication environment. Further, their presented utility function is just a simple weighted sum of values of various attributes, without considering other involved factors. [8] have presented a multi-attribute negotiation framework based on multi-agent systems for large-scale construction projects supply chain coordination. They have regarded supply chain as a typical multi-agent system. But, the model for utility determination presented by them represents the target utility in the form of other type of utility values. Their model can be helpful in the utility determination, but does not seem to provide concrete results for the target utility. They have presented the target utility, TU, as: \( TU = U_{BOT} + CS \), where \( U_{BOT} \) is the utility of own decision-making and Concession Step (CS) is determined by:

\[
CS = \beta(1 - \mu / U_{BOW})(U_{BOT} - U_{BOW}),
\]

where \( U_{BOT} \) is the utility of other participant’s decision-making, \( \mu \) is the minimal utility, and \( \beta \) is the negotiation speed. But, no discussion has been found on the computation of \( U_{BOT} \) and \( U_{BOW} \). [2] have presented a multi-attribute negotiation approach and utility model especially suited for the telecommunication domain. But, their presented utility function does not seem to calculate the concrete final value of utility. They have represented the utility in the form of other utility values. In their work, the total utility of a service combination, S, has been represented by following equation:

\[
u(S) = \sum_{c} k_i u_c,
\]

where \( k_i \) is the weight of a content-section and \( u_c \) is the utility associated with a content-section. Further, the \( u_c \) has been computed as the weighted-sum of the utilities of constituent medias, \( u_m(q_m) \), by following equation:

\[
u_c = \sum_{m \in M} \rho_m u_m(q_m),
\]

where \( \rho_m \) is the weight of a medium m. But, no discussion has been found on the computation of \( u_m(q_m) \). [11] have proposed a multi-attribute negotiation model with incomplete information. They also have presented a time-dependent negotiation strategy. In this strategy, a formulation for the utility that an agent desires to get in a time-period has also been presented. But, their presented utility function does not seem to calculate the concrete final value of utility, as they have represented the utility in the form of other utility-values. They have proposed the following relation:

\[
s_i(t) = 1 - (1 - ru_i) \left( \frac{t}{T_i} \right)^{\frac{1}{\beta}}
\]

Where \( s_i(t) \) is the utility that agent \( N_i \) desires to get in the current period t, 1 is the maximal utility agent \( N_i \) can get from the negotiation, \( ru_i \) represents the ultimate reservation utility of agent \( N_i \) for this
negotiation, $T_i$ is the deadline of agent $N_i$, and $\beta_i$ represents the strategy parameter of agent $N_i$.

It can be easily seen from this relation that the utility $s_i(t)$ has been represented in the form of other utility value $ru_i$, but no formulation has been found corresponding to $ru_i$.

The work presented in this paper tries to fulfill some of the shortcomings enumerated above. The work presents a utility based, multi-attribute negotiation model for negotiation between SWSs. The proposed work has presented a communication model for the negotiation between SR and SP using FIPA Communicative Acts [20]. The step-wise-step description of the negotiation process along with the algorithms for various activities has been presented. Further, the presented negotiation model proposes a feedback system by presenting a new data-structure, agreement table. It can expedite the negotiation process by reaching the agreement in lesser number of negotiation-steps. Hence, the presented negotiation approach for SWSs is more reliable, can provide more accurate decision-making, can fasten the process, and is more in line with the practical manual negotiation process.

Figure 8: Negotiation Agreements with various SPs

We have implemented a system for the problem of travel-booking providing negotiation between SWSs using proposed negotiation approach. The problem involves the booking of a flight for organizing a trip between two cities. The process consists of firstly discovering the potentials SPs which can provide the services for booking the flight between the required stations, after that the negotiation process starts with
the discovered SPs. The implemented system has used the proposed negotiation approach for the negotiation process. Figure 8 shows the result of negotiation with various SPs.

6. Conclusion
In this paper, mainly a utility based negotiation approach for negotiation between semantic web services has been presented. Along with the communication model and algorithms for various activities in negotiation process, the paper also proposes a negotiation feedback system. The feedback-system can expedite the negotiation process by decreasing the number of negotiation-steps in which agreement is reached. Based upon the proposed models, a prototype system providing negotiation between semantic web services has been implemented. The work has also been evaluated by comparing it against the existing similar works. Our future works involve enhancing further the proposed negotiation-approach.

References
(SDM '04) in conjunction with 30th International Conference on Very Large Databases.

[17] Kumar, S., Mastorakis, N. E., A Utility Model for Negotiation between Semantic Web Services, 8th WSEAS Int. Conf. on Telecommunications and Informatics (TELE-INFO’09), Turkey, 2009.


