A Context-Aware Approach For Service Selection Using Ontologies *

Murat Şensoy
murat.sensoy@boun.edu.tr

Pınar Yolum
pinar.yolum@boun.edu.tr

Department of Computer Engineering
Boğaziçi University
Bebek, Istanbul,Turkey

ABSTRACT

Selecting the right parties to interact with is a fundamental problem in open and dynamic environments. The problem is exemplified when the number of interacting parties is high and the parties' reasons for selecting others vary. We examine the problem of service selection in an e-commerce setting where consumer agents cooperate to identify service providers that would satisfy their service needs the most. Previous approaches to service selection are based on capturing and exchanging the ratings of consumers to providers. Contrary to previous, rating-based service selection, this paper advocates an objective experience-based approach for service provider selection, in which consumers record their *experiences* with service providers rather than the overall, subjective ratings for a provider. A consumer's experience with a service provider is represented using an ontology that can capture subtle details including the context in which the service was requested. When a service consumer decides to share her experiences with a second service consumer, the receiving consumer evaluates the experience using its own context and evaluation criteria. By sharing experiences rather than ratings, the service consumers can model service providers more accurately and thus can select the service providers for their needs more correctly.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Design, Experimentation

Keywords

Service Selection, Service Ontology

*We thank to the anonymous reviewers for helpful comments. This research is supported by Bogazici University Research Fund under grant BAP05A104.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

1. INTRODUCTION

Selecting partners to cooperate with is a challenging task in open environments, where the participants are autonomous. Open systems are not operated by a central authority that can monitor all agents' activities and ensure that everyone acts in the best interest of others. This implies that for a given service description, a plethora of service providers with substantially different service offerings will exist. A service consumer that is interested in receiving a particular service should then need to select a subset of the service providers that will satisfy her service needs in the best way.

Previous approaches have studied service selection by considering reputation and trust in agent societies. *Reputation systems* enable consumers to rate the service providers in a centralized location. The ratings of the consumers are then aggregated to decide whether a service provider will act as expected [12]. E-bay [1] uses such a reputation system. Whereas in closed settings a central authority (such as the company itself) exists to collect and aggregate ratings, there is no such authority for open systems. Hence the reputation systems are not directly applicable in open systems [16].

Distributed approaches to service selection consider *trust* among entities [17, 16, 9]. Trust captures a truster's expectation from a trustee for a particular service. Most formalizations of trust lack expressiveness and denote trust merely as a rating. However, the episode that leads to the rating is important for understanding the rationale for the rating [5]. For example, a service consumer may give a low rating to a service provider who delivers a book two days late. If the delivery date is not significant for a second service consumer, the first service consumer's low rating will not be significant, either. Hence, consumers should not record and exchange subjective ratings that depend on context, but objective information on actual experiences that can be reevaluated by the recipient based on context.

Accordingly, this paper develops an approach for distributed service selection that allows consumers to capture their experiences with the service providers using ontologies. The ontology used captures the requested service description and the received service in detail. The consumers can then exchange their detailed experiences of service providers rather than plain ratings. A consumer that receives another agent's particular experience evaluates the received experiences individually, considering her own context to decide on which service provider to select. Whereas rating-based approaches reflect the subjective opinion of the raters, the experience-based approach allows the objective facts of the experience to be communicated to the other party.

The rest of this paper is organized as follows. Section 2 explains representation of experiences using an ontology. Section 3 briefly describes our protocol for capturing and exchanging expe-

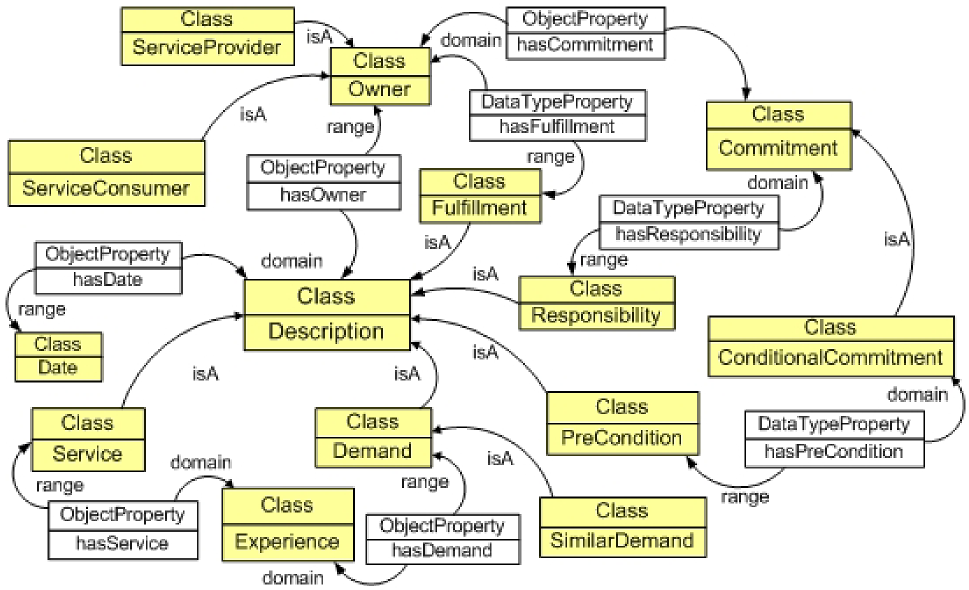

Figure 1: Base level ontology.

riences. Section 4 discusses how service consumers can reason about the experiences of others. Section 5 experimentally evaluates our experience-based service selection with comparisons to rating-based approaches. Section 6 discusses our work with references to the literature.

2. SHARING EXPERIENCES

We consider an architecture where service consumers are looking for service providers to handle their service demands. A *service demand* is expressed in terms of well-defined constraints on attributes of a service such as service completion time, service price, and so on. If a given service does not meet those constraints, then we assume that the owner of the demand be *unsatisfied*. However, another service consumer having weaker demand constraints could potentially be *satisfied*. If consumers only expose their levels of satisfaction (e.g., with a rating), the former service consumer will reveal a low level of satisfaction to the latter consumer. Even though the latter service consumer might have been satisfied with the service provider, she will not choose to interact with the service provider. Instead, if the latter consumer can use the *objective experience* data of the former by knowing the requested and supplied service characteristics, then she can make an informed choice. Consequently, a service consumer can model the service provider using the objective experience data of herself and other service consumers. Example 1 demonstrates how modelling of service providers using actual experiences plays an important role in making right decisions. Model of a service provider may be almost the same for different service consumers but interpretation of the model depends on the specific service consumer, her demand characteristics, and her satisfaction criteria.

EXAMPLE 1. *Consider a book seller who is competent in urgent deliveries to many places except Iowa City. If a service consumer wants an urgent delivery to Iowa City, she can use actual experiences of other consumers to model the service providers and recognize that the service provider could not make timely deliveries to Iowa City and choose another seller instead. Experience ontology should then capture the delivery location to express such nuances in experiences.*

Using objective experience data is promising for the selection of service providers. However, the following challenges need to be addressed:

- How will service consumers express their service demands and experiences?
- How will the experiences of interest be collected by a service consumer?
- How will service consumers use experiences to model service providers and to make decisions?

Service consumers use a common experience ontology for the specified service domain. This ontology covers the fundamental concepts (such as demand, service, and experience), which exist in the base level ontology and domain specific concepts and properties, which exist in the domain level ontology. Using those concepts and properties, a service consumer can express her service demands and experiences.

2.1 Base Level Ontology

The base level ontology (Figure 1) covers domain independent concepts of the experience ontology. Base level ontology is centered around the *Experience* class, whose instances represent experiences of service consumers. This is motivated by the concept of experiences in real life. An experience is a combination of what a consumer request from a service provider and what the consumer receives at the end. Accordingly, in the ontology, an experience consists of a service demand and supplied service for the demand. For this purpose, *Demand* and *Service* classes are included in the base ontology. Both demand and supplied service concepts are descriptions of a service for a specific domain and hence share a number of properties. In order to represent domain specific descriptions, *Description* class is introduced in base level ontology and it is extended in domain level ontology. This class includes generic information to describe basic concepts in the ontology. For example, descriptive classes such as *Demand* and *Service* are sub-classes of *Description*. Domain-dependent properties of *Description* class can be used to describe service demands, supplied services, responsibilities and fulfillments of sides during transactions. These properties are shown in domain level ontology.

Each *Description* has an owner and a date. *Owner* class is used to represent owners of descriptions. For a demand, owner is a service consumer and for a service, owner is a service provider. *ServiceConsumer* and *ServiceProvider* classes are subclasses of *Owner* class and represent service consumers and service providers, respectively. *Date* class represents the date of description. For an instance of *Service*, *Date* refers to the date of supplied service, whereas, for an instance of *Demand*, it refers to the date on which the demand is created.

An owner may have commitments [15]. Commitments are used to capture the responsibilities of a debtor to a creditor. A commitment always has an instance of responsibility. This means that the owner of the commitment is responsible for the realization of conditions described in the responsibility instance. Example 2 demonstrates a simple responsibility instance. *Commitment* and *Responsibility* classes are used to express commitments and responsibilities respectively in the experience ontology. As a result of transactions, parties have fulfillments. Fulfillments are accomplishments of responsibilities and are denoted with the *Fulfillment* class. *Fulfillment* and *Responsibility* classes are subclasses of *Description*. Owners of responsibilities or fulfillments can be service consumers or providers depending on the context.

EXAMPLE 2. *If a service provider has a commitment and responsibility of the commitment has toLocation property referring to New York City and has hasShipmentCost property referring to 5$, this means that the service provider is responsible for the delivery of the goods to New York City with a shipment cost of 5$.*

Unlike commitments, conditional commitments have preconditions [15]. A conditional commitment $CC(X, Y, P, Q)$ denotes that if the precondition P is carried out by Y, X will be committed to carry out responsibility Q. In this definition, Y is the owner of the precondition and X is the owner of responsibility. *ConditionalCommitment* and *Precondition* classes are used in the ontology to specify conditional commitments and preconditions. *ConditionalCommitment* class is a subclass of *Commitment* and *Precondition* class is subclass of *Description*. Conditional commitments can be used to represent contracts and offers made by service consumers and providers. An example case is demonstrated in Example 3.

EXAMPLE 3. *If the provider makes the shipment one week early, the consumer is committed to pay 1100$ for a product whose actual value is only 1000$. Service providers can also make offers using conditional commitments. The ontology can represent such contracts in demands as well as in offers.*

A service consumer may want to communicate with other service consumers with similar demands. But similarity is a subjective concept and may change for each consumer. To allow a consumer to express her description of a similar demand, *SimilarDemand* class is included in the experience ontology. *SimilarDemand* class is a subclass of *Demand*. A service consumer can express what a similar demand is with respect to her similarity criteria using Semantic Web Rule Language (SWRL) [2]. That is, if a consumer has a particular service demand and a list of others' experiences, then she can apply the SWRL rule to select those experiences in which the service demands were similar to that of her own. If the consumer makes her SWRL rule for similar demands public, other consumers can also use this expression of similarity to reason about whether their past service demands were similar to the demand of the consumer or not.

```
<ruleml:imp>
  <ruleml:_head><swrlx:classAtom>
      <owlx:Class owlx:name="#similarDemand" />
      <ruleml:var>DEMAND</ruleml:var>
    </swrlx:classAtom></ruleml:_head>
  <ruleml:_body>
    <swrlx:datavaluedPropertyAtom swrlx:property="#isRefoundable">
    <ruleml:var>DEMAND</ruleml:var><ruleml:var>Refoundable</ruleml:var>
    </swrlx:datavaluedPropertyAtom>
    <swrlx:datavaluedPropertyAtom swrlx:property="#hasDeliveryDuration">
      <ruleml:var>DEMAND</ruleml:var>
      <ruleml:var>DeliveryDuration</ruleml:var>
    </swrlx:datavaluedPropertyAtom>
    <swrlx:predicateAtom swrlx:predicate="http://kaon2...#ifTrue">
      <owlx:DataValue owlx:datatype="http://...XMLSchema#string">
          $1 && $2 <= 14</owlx:DataValue>
      <ruleml:var>Refoundable</ruleml:var>
      <ruleml:var>DeliveryDuration</ruleml:var>
    </swrlx:predicateAtom>
  </ruleml:_body>
</ruleml:imp>
```

Figure 2: Example KAON2 rule for similar demands

We use KAON2 [3] to perform the mentioned reasoning on experiences. In addition to OWL predicates, KAON2 supports some useful non-OWL predicates such as *ifTrue* and *evaluate* to test and evaluate logical and mathematical expressions. A simple rule for similarity is shown in Figure 2. In this rule, the consumer states that a demand for a service is similar demand only if the demand requires a refundable service in addition to a delivery duration less than or equal to 14 days. Unlike this example, rules for similarity can contain complicated logical statements about demand properties and conditional commitments.

2.2 Domain Level Ontology

Domain level ontology captures domain specific properties and concepts. Core class of domain level ontology is *Description* class. Domain specific properties of *Description* class is used to describe service demands, supplied services, responsibilities and fulfillments of parties during transactions. A domain level ontology for online shopping is shown in Figure 3. Properties of Description class in this ontology are *hasShoppingItem*, *toLocation*, *hasDeliveryType*, *hasDeliveryDuration*, *hasShipmentCost*, and *hasPrice*. These properties refer to shopping items, delivery location, delivery type, delivery duration, shipment cost and price respectively. Some boolean properties are also included in this set of properties: *isRefundable* and *hasConsumerSupport*. These properties indicate whether the transaction is refundable or not and whether the service provider supplies consumer support or not. The range of *hasShoppingItem* property is *ShoppingItem* class. This class has properties, *hasQuantity*, *hasUnitPrice* and *hasQuality*. The range of *hasQuality* property is *Quality* class. This class describes quality properties of shopping items. The properties of *Quality* class are not depicted because of space restrictions.

The properties of *Description* class also have slightly different meanings for different concepts. For example, *hasPrice* property refers to the money a consumer is willing to pay for a service if it is used to describe a service demand. This property refers to the money the consumer is requested to pay for the supplied service if it is used to describe provided service. If *hasPrice* is used to describe a responsibility, *hasPrice* refers to the money a service consumer promises to pay for a service or it refers to the money a service provider will accept for a service, depending on the owner of the responsibility. For the first case, service consumer is the owner of responsibility and for the second case owner is service provider. If

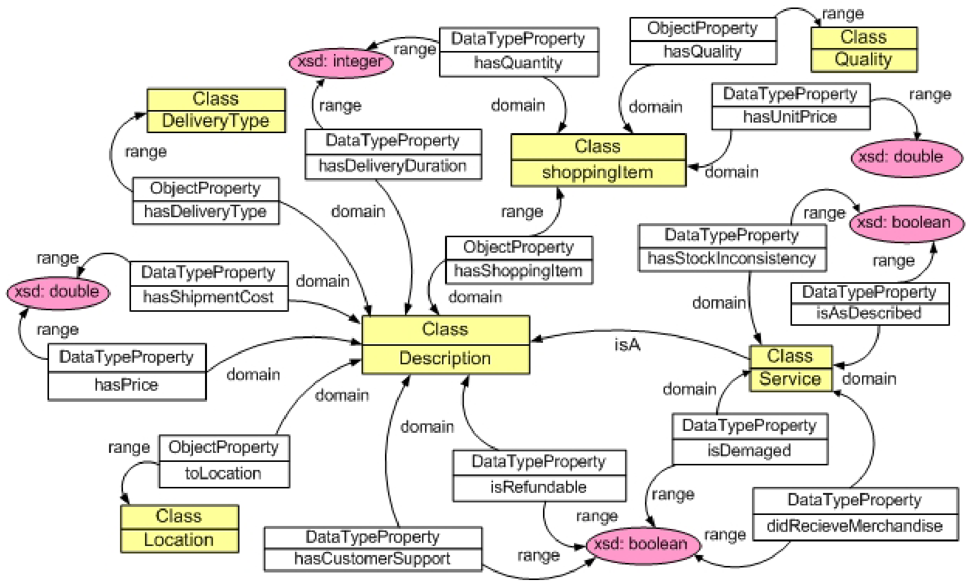

Figure 3: Domain level ontology for online shopping.

hasPrice is used to describe a fulfillment of a service consumer, *hasPrice* refers to the money paid by the service consumer for the specified service.

In addition to the properties of *Description* class, concepts in the ontology may also have domain-specific properties that other concepts do not have. For example, for consumer goods domain, properties such as *didRecieveMerchandise*, *hasStockInconsistency*, *isAsDescribed* and *isDemaged* are included as properties of *Service* class in domain level ontology. The property *didRecieveMerchandise* denotes whether the merchandise is received by the consumer or not. The *hasStockInconsistency* property denotes if the provider claimed to have the product in stock even though it was not. Other properties can be added to this ontology.

3. BUILDING A CONSUMER SOCIETY

A consumer society emerges as a result of consumers' need to retrieve experiences of other consumers. Each service consumer in the society is represented with an agent that is uniquely identified (possibly by an IPv6 address). Each service consumer knows only a subset of all consumers in the society and lists these consumers in an *acquaintance list*. An acquaintance list is a dynamic list of service consumers having service demands classified as similar demand by the owner of the list. When a new service consumer joins the society, its acquaintance list is populated with a small number of randomly chosen service consumers. Acquaintance lists are not symmetric: X may be on Y's list but that does not mean that Y will be on X's list. Each consumer collects others' experiences in an *experience repository*. Each time a service consumer makes a decision, it uses the experiences in this repository. The service consumer refreshes and updates its experience repository periodically by removing old experiences and adding new experiences.

Initially, service consumers do not have any experiences. When a service consumer X needs experiences for reasoning on which service provider to choose, it should discover other service consumers having similar demands and should populate its acquaintance list with those service consumers and their service demands. In real situations, consumers usually change their demand characteristics for the same service domain. Example 4 illustrates such a case. Hence, it may be necessary to collect relevant experiences periodically. In order to accomplish this, a consumer follows the procedure summarized in Algorithm 1 [6].

EXAMPLE 4. *A consumer living in Istanbul orders a best seller for himself and a second copy for his brother living in Iowa City. For the first book, the price is more important than delivery duration. However, the second book is a birthday present and must be delivered on time. For this order, delivery duration is more important than price for the consumer.*

Algorithm 1

```
1:  Decide Shopping
2:  if (Shopping) then
3:    Check Experience Repository
4:    while (Not Have Enough Experience) do
5:      Check Acquaintance List
6:      if (Not Have Enough Acquaintance) then
7:        Get New Acquaintances: Using PDM or RAM
8:      end if
9:      Get Experiences: Using REM
10:   end while
11:   Select Provider
12: end if
```

When a service consumer decides to receive a service (Line 1), it checks its experience repository. In order to make reliable decisions, the service consumer should compute the minimum number of experiences for decision making within a 99% confidence interval [11]. If the number of experiences in the repository is not enough, the service consumer collects new experiences. However, in order to collect new experiences, the consumer should have sufficient number of acquaintances. If it does not have, it should increase the number of its acquaintances (Lines 5, 6).

To increase its acquaintances, a consumer can use two messages: *Peer Discovery Message (PDM)* and *Request for Acquaintances Message (RAM)*. Both PDM and RAM messages contain an SWRL rule that expresses the similar demand criteria of the originator of the message. When a consumer Y receives a PDM message, it checks if its service demands are similar to that of the originator X. If so, it notifies X and X adds Y as a new acquaintance entry in its acquaintance list. This entry contains identity of Y and its demands classified as similar demand by similarity criteria of X. The consumer Y also forwards the request to a set of service consumers in its acquaintance list. Y selects consumers having demands similar to demand of the originator to forward the request. If there is no such consumer, Y randomly selects consumers from its acquaintance list. How long the request is going to be forwarded is controlled using a time-to-live field. All other agents that receive the request act the same way Y does. When Y receives a RAM message from the originator X, it checks its acquaintance list for entries containing consumers having demands similar to the demand of X. Then Y sends these entries to X and X can add these entries to its acquaintance list.

The service consumer populates it acquaintance list through PDM and RAM messages. After having sufficient number of acquaintances, the consumer uses *Request for Experience Message (REM)* to collect new experiences (Line 9). Interestingly, an REM message also contains a rule for expressing similar demand criteria of the sender. When service consumer Y gets an REM message from service consumer X, it evaluates its demands in its experiences using the similarity criteria in the REM. Later, it can send its experiences to X if the experiences have similar demands with respect to similarity criteria encapsulated in the REM, so that X can populate its experience repository with these experiences. After collecting sufficient number of experiences, X uses experiences in its repository for decision making.

4. MODELLING SERVICE PROVIDERS

When a service consumer decides to receive a service, it models each service provider using the experience data available in its repository and selects a provider with the highest probability to satisfy its needs. In determining a suitable provider, the consumer uses information contained in actual experience data. The consumer can make use of experiences by giving them different weights: for example, it can give more weights to its own experiences than the experiences of others. Demand and service specifications within experiences are received in the form of ontologies, but then they can be converted into internal representation of the service consumer in order to speed up decision making.

For modelling service providers, aggregation of their past service offerings is used. To aggregate the experience instances, agents can use various methods. The method used in the simulations is parametric classification. In this method, the service consumer models each service provider and builds a multidimensional Gaussian model for each service provider using the collected experience data related to that provider. There are two classes for each model: satisfied and dissatisfied.

Demand and commitment information in each experience is represented as a vector. Each field in this vector is extracted from the experience ontology. These fields may correspond to property values in experience ontology such as service price. Then, supplied service for this demand is classified as satisfied or dissatisfied with respect to satisfaction criteria of the consumer. The (vector, class) pairs are used as training set.

For each class, the variance and the mean are extracted from the training set. Those parameters are used for modelling of the class. Then, for each of the classes, a discriminant function is defined to compute the probability of satisfaction [8]. The service consumer performs this computation for every service provider and chooses the provider with the highest satisfaction probability.

5. SIMULATIONS

In order to demonstrate the performance of the proposed methods, we implemented a simulator and conducted simulations on it. The main purpose of these simulations is to see the effect of our model on the selection of an appropriate service provider for a service consumer. In the simulator, two types of service provider selection strategies are implemented and compared with each other in terms of achieved satisfaction. Those strategies are explained below.

- **Service Provider Selection Using Experiences (SPS_{EXP}):** This strategy uses the proposed method for service provider selection. Experiences related to service demand of an agent are collected by the agent and decisions are made using those experiences.

- **Service Provider Selection Using Ratings ($SPS_{ratings}$):** For a new service demand, a service consumer agent can select a service provider using ratings from other consumer agents. In this strategy, in order to make better satisfaction-targeted selections, ratings are taken from those agents who have similar demands with respect to similarity criteria of the agent. So, both SPS_{EXP} and $SPS_{ratings}$ actually use the information from the same service consumers for a given decision process. $SPS_{ratings}$ uses ratings while SPS_{EXP} uses experiences.

To be more focused, the simulations enforce agents to make decisions based on others' experiences rather than their previous experiences. As is the case with real world, service consumers periodically change their service demands. This is done to allow variations on context such as the case shown in Example 1.

5.1 Simulation Environment and Settings

The mentioned simulator is implemented in Java and KAON2 is used as OWL-DL reasoner [3]. Simulations are run on an IBM server with 2.8 GHz CPU and 4.0 GB RAM under Windows 2003 server OS. Simulations are repeated 10 times in order to increase the reliability.

In the simulations, initially there are no experiences accumulated in the agent society yet. As time goes by, the number of experiences increases. During decisions, each agent uses only the experiences of others. Each service consumer changes its demand characteristics after receiving a service with a predefined probability denoted as P_{CD} and collects experiences for its new demand. Each service consumer has a probability of requesting a service for any epoch. This probability is uniformly chosen between 0 and 1. In other words, only around 50% of consumers consume service at a given epoch. An online shopping market with 20 service providers and 400 service consumers are simulated in the simulations. Simulations are run for 100 epochs. Experiences expire after 20 epochs to keep experience repositories fresh and small. Simulations are also conducted for other settings, because of space limitations they are not shown here. Simulation results for other settings are comparable to results shown in Section 5.2.

Note that service consumer X may find the demand of service consumer Y similar to its own, but the service provider that satisfies the demand of Y may not satisfy the demand of X. This could be the case because the satisfaction criteria of Y may be highly different or conflicting with the satisfaction criteria of X. This fact is also imitated in the simulations. That is, a fraction of the service consumers who have the similar service demand to X have a satisfaction criteria that is different from the satisfaction criteria of

X. This fraction is denoted with β. The remaining consumers have satisfaction criteria identical to that of X.

In the simulations, service characteristics of a service provider are generated as follows. Each property of *Service* class represents a dimension in a multidimensional service space. Dimensions of service space used in the simulations are tabulated in Table 1. Each service provider has a multi-dimensional region called service region in this service space. This region is randomly generated. Service space and service regions have 15 dimensions. Service region covers all of the services produced by the service provider.

EXAMPLE 5. *If a consumer located in Istanbul orders two books titled Anagrams from a service provider, then the service that the provider delivers will be constructed as follows: The properties that are specified (shopping item, quantity and location) will be fixed. For the remaining attributes, the service provider will choose random values making sure that the values stay in the range of its service region. So, for this example, the degree of freedom for generating services will be reduced to* 12*.*

Service description of service providers is enhanced by adding conditional commitments to service regions. Conditional commitments are produced using functions of service space dimensions. Example 6 depicts a production of conditional commitment. Similar conditional commitment functions are generated between other service attributes.

EXAMPLE 6. *If one book is bought, unit price will be 12$ and if four books are bought, unit price will be reduced to 6$. Such conditional commitments are represented using multi-dimensional functions. For the previous example, a non-linear function of quantity and unit price is used, ($UnitPrice = 8/Quantity + 4$).*

Demand of a service consumer is generated as the following. Demand space is constructed by removing dimensions of service space, which do not belong to *Demand* class. Then, a region named demand region is chosen randomly. Center of this region represents the demanded service. If provided service for this demand stays within the margins of demand region, the service consumer having this demand gets satisfied, otherwise she gets dissatisfied. If the shipped products are damaged or merchandise is not received, the consumer also gets dissatisfied. Demands are generated so that each demand will be satisfied by at least one service provider.

Similar demand criteria for the service consumer is created as the following. A new region named *similar demand region* is constructed by removing some dimensions of the demand region. Number of dimension to be removed and these dimensions are chosen randomly. Service demands staying within the margins of the *similar demand region* is classified as similar demand by the consumer. In simulations, experience ontology is used to describe experiences and similar demand specifications as explained in Section 2.

Table 1: Dimensions of service space and their ranges

Dimension Name	Type	Range
hasShoppingItem	Integer	1 - 1000
toLocation	Integer	1 - 100
hasDeliveryType	Integer	1 - 6
hasDeliveryDuration	Integer	1 - 60
hasShipmentCost	Double	0 - 250
hasPrice	Double	10 - 11000
hasUnitPrice	Double	1 - 100
hasQuantity	Integer	1 - 100
hasQuality	Integer	1 - 10
isRefundable	Boolean	0 - 1
hasConsumerSupport	Boolean	0 - 1
didRecieveMerchandise	Boolean	0 - 1
hasStockInconsistency	Boolean	0 - 1
isAsDescribed	Boolean	0 - 1
isDemaged	Boolean	0 - 1

5.2 Simulation Results

This section summarizes the results of the simulations. There are two primary parameters in the simulations: P_{CD} and β. Let X be a service consumer making service provider selection, and S be the set of service consumers having demands similar to the demand of X with respect to similarity criteria of X at the time X makes a decision. When X is using $SPS_{ratings}$ for decision making, it takes ratings only from consumers in the set S and X takes experiences from these service consumers while it is using SPS_{EXP} for decision making.

Figure 4 shows the simulation results for P_{CD}=0, β=0.5. This is the case when the consumers do not change their service demands and half of the consumers in set S will have satisfaction criteria conflicting with the satisfaction criteria of X.

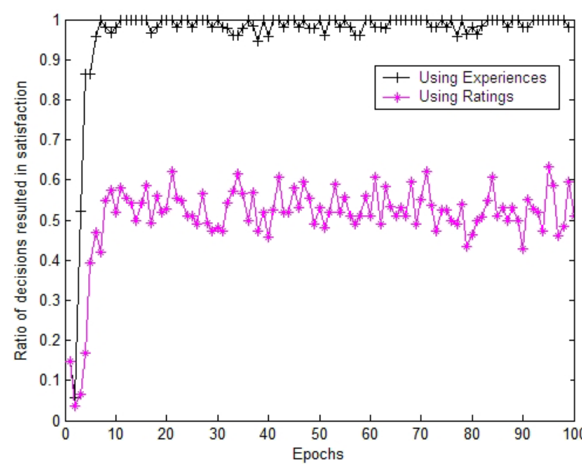

Figure 4: Simulation results for the settings: 20 service providers, 400 service consumers, P_{CD}=0, β=0.5.

When this is the case, only 50% of the services lead to satisfaction of service consumers if these service consumers use $SPS_{ratings}$. On the other hand, service consumers using SPS_{EXP} are almost always satisfied with the supplied service. This is because only half of the ratings taken from the consumers in S lead to decisions that result in satisfaction on the average. However, the other half of the ratings lead to dissatisfaction.

However, as was illustrated in Example 4, consumers change their demand based on circumstances. Figure 5 shows simulation results for parameters P_{CD}=0.2 and β=0. This means that a consumer will change its current service demand with a probability of 0.2 after receiving a service. Figure 5 indicates that the performance of $SPS_{ratings}$ decreases seriously with time, where as the performance of SPS_{EXP} is constant around 100% satisfaction. Since the ratings of a consumer will be aggregation of its past transactions for all of its previous demands, as the consumers change their demands, their ratings will be more misleading than before. However, SPS_{EXP} uses actual experience data, which is free of any evaluation and uses this information for building models of service providers. Hence, as seen in the Figure 5, SPS_{EXP} leads to decisions with 100% satisfaction but satisfaction ratio of

$SPS_{ratings}$ decreases as consumers change their demands.

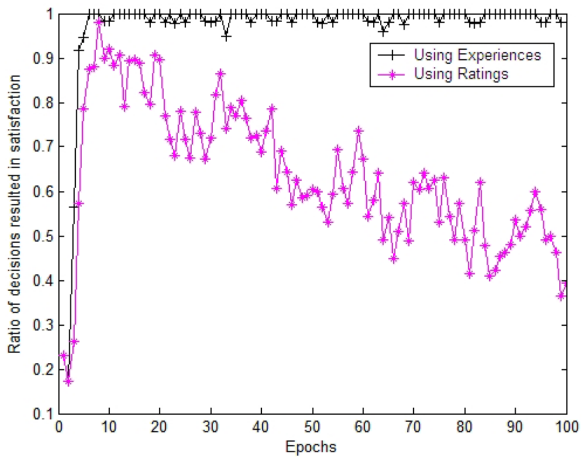

Figure 5: Simulation results for the settings: 20 service providers, 400 service consumers, P_{CD}=0.2, β=0.

Figure 6 shows simulation results for parameters P_{CD}=0.2 and β=0.5. This shows that the consumers will change their service demand with a probability of 0.2 and half of the consumers with the same demand will have conflicting satisfaction criteria. Performance of $SPS_{ratings}$ decreases further for these settings. However, for SPS_{EXP}, satisfaction ratio is around 1 after 5^{th} epoch (before 5^{th} epoch, there are not enough experiences accumulated in the environment for the modelling of service providers). This means that proposed method almost always leads to satisfaction of consumers. In order to see influences of β and P_{CD} on the satisfaction ratios achieved by SPS_{EXP} and $SPS_{ratings}$ strategies, simulations are repeated for different β and P_{CD} values.

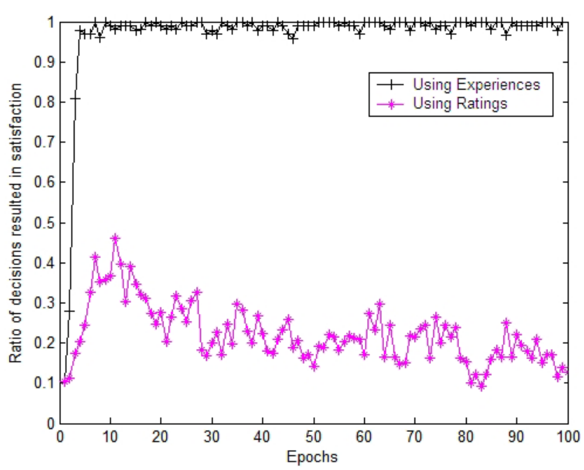

Figure 6: Simulation results for the settings: 20 service providers, 400 service consumers, P_{CD}=0.2, β=0.5.

Average ratio of satisfactions for these simulations are shown in Table 2 and Table 3. Performance of $SPS_{ratings}$ decreases considerably with an increase in the value of β or P_{CD}. These parameters are independent of each other. So, combinatorial effect of these parameters on the performance of $SPS_{ratings}$ is the multiplication of influences of each parameter. Simulations show that proposed method, SPS_{EXP} is robust to changes in β and P_{CD} parameters. Unlike $SPS_{ratings}$, performance of SPS_{EXP} does not change with changing β and P_{CD} values as well as the achieved satisfaction is 100% if service consumers use SPS_{EXP} for decision making.

Table 2: Average ratio of satisfaction with respect to different β values (P_{CD} is set to 0).

β	0	0.1	0.2	0.3	0.4	0.5
SPS_{EXP}	0.97	0.97	0.975	0.98	0.98	0.98
$SPS_{ratings}$	0.95	0.85	0.76	0.68	0.61	0.53

Table 3: Average ratio of satisfaction with respect to different β and P_{CD} values.

P_{CD}	SPS_{EXP}		$SPS_{ratings}$	
	$\beta = 0$	$\beta = 0.5$	$\beta = 0$	$\beta = 0.5$
0.0	0.97	0.97	0.95	0.53
0.1	0.97	0.96	0.72	0.35
0.2	0.97	0.97	0.56	0.28
0.4	0.98	0.98	0.42	0.18
0.6	0.98	0.97	0.37	0.15
0.8	0.97	0.98	0.32	0.13
1.0	0.98	0.98	0.28	0.12

For the setting β=0 and P_{CD}=0, satisfaction ratio of the strategy $SPS_{ratings}$ approaches 1. However, this setting is far from reflecting the real world. If a consumer classifies the demand of another consumer as similar demand, any provider satisfying the latter will also satisfy the former in case of β=0. This means that satisfaction criteria of the latter is identical to satisfaction criteria of the former. Therefore, ratings will be adequate for the right selection.

6. DISCUSSION

Previous approaches for service provider selection are mostly based on ratings that are inherently subjective. Instead, we are proposing to represent the steps that result in the rating, such as the requested service, the received service, and so on explicitly. This enables the collected information to be reinterpreted based on context. The performed simulations show that the use of experiences improves the decisions of the service consumers and increase the overall satisfaction significantly compared with the rating-based service selection strategy.

Current service provider selection strategies accept ratings as first-class citizens, but do not allow more expressive representations like we have here. Whereas rating-based approaches assume that the ratings are given and taken in similar contexts (e.g., in response to similar service demand), we can make the context explicit. This allows agents to evaluate others' experiences based on their needs. Thus, the use of experiences improves the satisfaction ratio of the consumers.

Different AI techniques, such as case-based reasoning (CBR) can also be applied on objective experience data to select service providers. In [7], we propose a CBR approach for service selection and compare it with the parametric classification. In the CBR approach, a service consumer collects the most similar experiences from other service consumers. Then, the consumer evaluates the supplied services within these experiences using its satisfaction criteria and selects the service provider which provides the most satisfactory service. When service providers do not change the quality of their services, both reasoning techniques perform equally well in finding service providers. However, CBR approach finds the service providers in a shorter time than the parametric classification. On the other hand, if the service providers vary their service offerings even a small percentage, then the CBR approach performs much worse than the parametric classification.

FIRE [9] is a trust and reputation model consisting of four components: interaction trust, role-based trust, witness reputation and certified reputation. Witness reputation component is directly related to our approach since it allows agents to locate others by making use of other agents' past experiences. However, in FIRE the past experiences are captured only as ratings. However, in our approach, agents exchange their experiences in the form of ontologies so that they can represent their demands, received services, and so on in more depth.

Sen and Sajja [13] develop a reputation-based trust model that is used for selecting processor agents for processor tasks. Each processor agent can vary its performance over time. Agents are looking for processor agents to send their tasks to using only evidence from others. Sen and Sajja propose a probabilistic algorithm to guarantee finding a trustworthy processor. In our framework, service demands among agents are not equivalent; hence a provider that is trustworthy for a consumer need not be so for a different consumer.

Yolum and Singh study properties of referral networks for service selection, where referrals are used among the service consumers to locate the service providers [16]. Current applications of referral networks also rely on exchanging ratings. Hence, suffer from circulation of subjective information. It would be interesting to combine referral networks with the ontology representation here so that agents can exploit the power of ontologies for knowledge representation as well as referrals for accurate routing.

Zhang *et. al.* propose a multiagent approach for distributed information retrieval task [18]. In their work, each agent has a view of its environment called agent-view. The agent-view structure of an agent contains information about the language models of documents owned by each agent. An agent-view reorganization algorithm is run to dynamically reorganize the underlying agent-view topology. An agent-view is analogous to acquaintance list structure in our work. Zhang *et al.*'s protocol does not use ontologies or description logic reasoners during information retrieval. However, if their protocol is modified to accommodate the experience ontology and DL reasoners, their protocol can be used for retrieving experiences instead of the protocol that we have proposed in Section 3.

Soh and Chen propose a multiagent approach to improve distributed information retrieval performance by balancing ontological and operational factors [14]. In this work, collaborating agents enhance their performance by learning ontologically and operationally. Soh and Chen show that their proposed approach is able to improve the quality of the collaborations in terms of the response time, quality of the retrieved results, the number of neighbors contacted and message complexity. A similar approach can also be applied to our work in order to improve quality of collaboration.

Maximilien and Singh develop a QoS ontology to represent the quality levels of service agents and the preferences of the consumers [10]. Their representation of QoS attributes is richer (such as availability, capacity, and so on), however, their ontology does not represent commitments and thus business contract as part of the ontology. Further, their system does not allow reasoning by agents individually as we have developed here.

As future work, we plan to study the role of contracts in service selection. Further, we plan to extend our ontology using OWL-S [4] to make our approach applicable for Web services.

7. REFERENCES

[1] E-bay web site: http://www.ebay.com, 1995.

[2] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004.

[3] Kaon2 web site: http://kaon2.semanticweb.org, 2005.

[4] Owl-s: Semantic markup for web services, 2005.

[5] C. Castelfranchi and R. Falcone. Principles of trust for MAS: cognitive anatomy,social importance, and quantification. In *Proceedings of the 3rd International Conference on MultiAgent Systems*, pages 72–79, 1998.

[6] M. Şensoy, F. C. Pembe, H. Zırtıloğlu, P. Yolum, and A. Bener. Experience-Based Service Provider Selection in Agent Mediated E-Commerce. *To appear in Engineering Applications of Artificial Intelligence*, 2006. Available at: www.cmpe.boun.edu.tr/˜pyolum/eaai-2005.pdf.

[7] M. Şensoy and P. Yolum. A comparative study of reasoning techniques for service selection. Technical Report FBE(CmpE01)2006-2, Boğaziçi University, 2006. Available at: www.cmpe.boun.edu.tr/˜pyolum/SensoyYolum-TechReport.pdf.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, West Sussex, 2001.

[9] T. D. Huynh, N. R. Jennings, and N. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of 16th European Conference on Artificial Intelligence*, pages 18–22, 2004.

[10] M. Maximilien and M. P. Singh. A framework and ontology for dynamic web services selection. *IEEE Internet Computing*, 8(5):84–93, 2004.

[11] D. C. Montgomery. *Design and analysis of experiments*. John Wiley and Sons, West Sussex, 2001.

[12] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 475–482, 2002.

[13] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 288–293, 2002.

[14] L.-K. Soh and C. Chen. Balancing ontological and operational factors in refining multiagent neighborhoods. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 745–752, 2005.

[15] P. Yolum and M. P. Singh. Flexible protocol specification and execution: Applying event calculus planning using commitments. In *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 527–534, 2002.

[16] P. Yolum and M. P. Singh. Engineering self-organizing referral networks for trustworthy service selection. *IEEE Transactions on Systems, Man, and Cybernetics*, A35(3):396–407, 2005.

[17] B. Yu and M. P. Singh. Emergence of agent-based referral networks. In *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1268–1269, 2002.

[18] H. Zhang, W. B. Croft, B. Levine, and V. Lesser. A multi-agent approach for peer-to-peer based information retrieval system. In *Proceedings of the Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 456–463, 2004.