International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 5 (2013) pp. 354-364

Aspect-oriented Trust Based Mobile Recommender System

Punam Bedi[1] and Sumit Kumar Agarwal[2]

[1, 2] Department of Computer Science, University of Delhi,
Delhi, India
punambedi@ieee.org, sumitsagarwal@rediffmail.com

Abstract: **With the rapid advancement of wireless technologies and mobile devices, service recommendations have become a crucial and important research area in mobile computing. Although various recommender systems have been developed to help users to deal with information overload, few systems focus on personalized trustworthy recommendation generation for mobile users. In real life, trust plays an important role in the decisions related to sources that are used by human to take recommendations. This paper presents Aspect-Oriented Trust Based Mobile Recommender System (AOTMRS) that uses the concept of trust and Aspect Oriented Programming for advice-seeking and decision-making process similar to real life. The proposed system AOTMRS builds a mobility aspect and generates the trustworthy recommendations based on the user preferences and his demographic information such as location, time, need etc. AOTMRS exploits two peculiar characteristics of mobile information services such as "location-awareness", i.e., the knowledge of the user's physical position at a particular time and "ubiquity", i.e., the ability to deliver the information and services to mobile users wherever they are, and whenever they need. Implementing user mobility in multi agent recommender system using conventional agent-oriented approach creates the problem of code scattering and code tangling. The mobility aspect separates the mobility crosscutting concerns, which in turn improves the system reusability, maintainability and removes the scattering and tangling problems of the recommender system. Moreover, because of resource limitations of mobile devices such as display size etc, AOTMRS uses personalized visual interface for mobile and computationally light algorithm at the user end in order to provide more effective and persuasive recommendations. The prototype of AOTMRS has been designed and developed for tourism recommendation system. Performance of the proposed system is evaluated using precision and recall metrics.**

Keywords: Aspect Oriented Programming (AOP), Recommender System, Trust, Multi-agent System (MAS), Intuitionistic Fuzzy Set, Mobility, Crosscutting.

1 Introduction

People are often overwhelmed with the number of options available to them. Recommender systems or the concept of automatic recommendation generation is one of the possible solutions of this information overload problem. Recommender Systems attempt to reduce information overload and retain customers by selecting a subset of items from a universal set based on user preferences [2]. Many recommendation applications are intrinsically related to physical locations, for example to restaurants, monuments, museums, amusement parks, theaters, cinemas, emergency facilities etc. Their interest for a mobile user is often dependent on his/her proximity. For example, a driver through the Death Valley may ask about the location of gas stations in order to minimize the risk of running out of fuel, or to minimize the expense. Hence user mobility is one of the core design issues of the recommender system to generate more relevant recommendations for such applications. In mobile environment, due to the limitations of mobile devices such as small display size or missing keyboard etc, data input and information browsing is inconvenient [24]. By minimizing the interaction (keystrokes) between mobile user and the recommender system, user experience can be improved. The proposed system AOTMRS provides personalized visual interface for mobile users, which substantially reduces users' interaction effort.

Mobile Recommender systems for tourism industry generally use the similarity between the user and recommenders or between the items for recommending list of places to visit [9]. These systems usually do not take into consideration the social trust network between the entities in the society to ensure that the user can trust the recommendations received from the system. Hence the system model does not match the mental model of the user. In real life, trust plays an important role in the decisions related to sources that are used by human to take recommendations. If recommender system also use advice-seeking and decision-making process similar to real life, then it will be easier for the user to trust the recommendations of the system. Hence, by adding a trust component to a decentralized environment, the problem of lack of trust on the recommenders is alleviated.

The proposed system AOTMRS incorporates social elements of decision making and advice seeking by adding the trust component with the recommendation process. The AOTMRS also uses the Intuitionistic Fuzzy Sets (IFS) to handle fuzziness and uncertainty that are inherent within the social recommendation process. The IFS [1] having degree of membership, degree of non-membership and degree of uncertainty are very well suited for modeling fuzziness and uncertainty in the recommender systems. Bedi et al. [4]

introduced trust based recommendation algorithm which allows agents in decision making to generate the recommendations based on their trustworthiness. This recommendation algorithm generates the recommendations based on explicit user preferences only. We enhanced a prior recommendation algorithm [4] by incorporating implicit user preferences i.e. contextual information such as user location, time etc within recommendation generation process. The enhanced work considers only those places as recommendations which are near of user's current location or the places where the user can reach from his current location within the time period explicitly specified by the user. The proposed system AOTMRS combines the explicit user preferences (type of place to visit, climate etc) through visual interface and implicit user preferences (user location, time etc) automatically for recommendation generation.

Research literature on intelligent agent system architectures has established that the problems that are inherently distributed can be efficiently implemented as a multi-agent system [2]. As the recommenders may be geographically distributed, this recommendation generation problem for the mobile users becomes a distributed computing problem. Hence the system AOTMRS proposes the solution based on multi agent system (MAS) approach, where every user in the system is represented by an agent. The agents within AOTMRS communicate with others to generate the recommendation. Implementation of user mobility in recommender system using conventional agent-oriented approach, relevant mobility specific code replicates and spreads across several agent class hierarchies. So adding or removing the mobility code requires invasive changes in these agent classes. Using conventional agent-oriented approach, even if we try to move mobility specific methods and attributes from the agent classes to a new class, the following problems still remain: (i) agent classes need to keep an attribute with a reference to this new mobility related class, and (ii) the code relative to identifying user location and his preferences remains scattered over the methods on other agent classes. As a result, mobility concern still crosscuts multiple agent class hierarchies representing other agent concerns, such as collaboration and agent's basic functionality. This has led us to work on how we can eliminate these mobility-specific crosscutting problems of the recommender system by using the concepts of AOP.

The main contribution of this research work are three-fold; First, this paper presents a solution of mobility-specific crosscutting problems of the recommender system by creating a mobility aspect that provides support to improve the system reusability, maintainability. This mobility aspect is the integration of type-based and policy-based mobility strategies that weaves with several agents of AOTMRS to effectively handle the scattering and tangling problems [21] of the recommender systems. Second, it presents improved trust based recommendation algorithm that matches the mental model of decision making of a human being in order to improve the relevance, usefulness and quality of the recommendations. Third, this research work presents a personalized visual interface for mobile users in order to substantially reduce users' interaction effort. The proposed visual interface for recommendations is interactive and is customized according to user's preferences. User can amend his current requirements interactively in order to obtain the improved recommendations from the Recommender System.

The rest of the paper is organized as follows. Section 2 reviews the related work in this area. Concepts used in our proposed approach are discussed in section 3. Section 4 presents our approach to deal with mobility crosscutting concerns in the recommender system along with the architecture of AOTMRS system. The proposed trust based recommendation algorithm is discussed in section 5. Description of proposed personalized visual interface is given in section 6. Experimental details are shown in section 7 and finally conclusion and future work are discussed in section 8.

2 Related Work

Ricci and Nguyen [19] introduced critique-based recommendation methodology to support product selection decisions for mobile users. Hussein [15] presented the enhanced K-means algorithm to minimize the validity (intra-cluster/inter-cluster) to get optimum number of clusters for generating most suitable recommendations for mobile user. Yang and Wang [23] proposed a location-based data and service middleware based on service-oriented architecture using GPS and Web2.0 Service in order to implement mobile information pushing system. They also presented 3-D Tag-Cloud module to visualize useful retrieval information even in the limited mobile screen. Yeung and Yang [24] developed a proactive personalized mobile news recommendation system by using the Bayesian network and Analytic Hierarchy Process (AHP) model to deal complex multi-criteria recommendation process. Tangphoklang et al. [20] presented mobile multi-criteria recommendation architecture for mobile banking service domain. This architecture was somewhat inflexible for distributed work flow to reach wider clients or alliance banking services. Tumas and Ricci [22] developed personalized mobile city transport advisory system using knowledge-based approach for providing best route between two arbitrary points in the city. Park et al. [18] proposed a probabilistic model using Bayesian network to identify the minimal set of parameters that are important for generating user preferences. Burigat et al. [9] introduced decision-support functions with a map interface to generate recommendations for visitors. Church and Smyth [11] proposed community-based personalized meta-search algorithm for result selection. Dunlop et al. [12] applied constraint-based filtering approach to compute personalized recommendations. Honda et al. [14] presented new user-item clustering method based on a structural balance theory for recommendation generation. Kiczales et al. [16] proposed the new program development methodology AOP to generate more efficient, modularize and better understandable code. They shown that program written in AOP is more easily maintainable. Mehmood at el. [17] developed a framework that separates the performance aspects from other agent hood, functional and non-functional aspects to improve modularity of MAS using AOP. Bedi at el. [5, 6, 7] explored various inherent problems and their side effects due to which crosscutting concerns cannot handle effectively in design of the multi-agent based recommender system.

Although the considerable amount of work has been done on user mobility issues of recommender systems and some research work focuses on personalized trustworthy recommendation generation and mobility crosscutting issues of recommender system, the problems of tangling and scattering in recommender systems has not been resolved as per our knowledge. We propose a novel approach AOTMRS which handles these mobility concerns in a better modular way to reduce the cognitive complexity of the recommender system using AOP and improves the usefulness and quality of the recommendations by incorporating the trust component within the recommendation process.

In this paper work we have developed the mobility aspect and integrated this mobility aspect with various agents of multi agent based recommender system using the concept and constructs of AOP such as: joinpoints, pointcuts, advice and aspect etc. The mobility aspect separates the mobility protocol from agent classes, such as agent types, plans and roles to identify the user location and his needs.

3 Background

This section briefly describes the concept of Recommender Systems and AOP.

3.1 Recommender Systems

Recommender system is a technological proxy used for searching out a good option amongst a potentially overwhelming set of alternative products or services. A recommender system recommends items to users by predicting items relevant to the user, based on user profile which contains various kinds of information including items, user information and interactions between users and items [2, 3]. Recommender systems use the opinions of members of a community to help individuals in that community, by identifying information most likely to be interesting to them or relevant to their needs. These systems form a specific type of information filtering technique that attempts to present information items such as movies, music, books, news, images, web pages, that are most likely to be of interest to the user. These systems are very powerful cognitive decision-makers in the context of distributed online information processing in real-time networked scenarios [2].

Recommender Systems have come up in the market to advice users so that the right information is delivered to the right person at the right time. These systems can be classified mainly into four categories: content based, collaborative filtering based, knowledge based and hybrid recommender systems.

- Content based recommender systems use the ratings of the items the target user liked in the past to predict items the target user would also like.
- Collaborative filtering based recommender systems use the ratings of the users with tastes “similar” to the target user and predict items for the target user.
- Knowledge based recommender systems use a knowledge structure to make inferences about the user needs and preferences.
- Hybrid recommender systems use a combination of the above approaches.

3.2 Aspect Oriented Programming (AOP)

Aspect Oriented Programming (AOP) is a programming paradigm which aims to improve modularity of the software systems by allowing separation of crosscutting concerns. Crosscutting concerns usually refer to non functional properties of software such as learning, security, transaction management, synchronization, mobility, and error handling. Implementation of crosscutting concerns in a recommender system by using a conventional agent oriented approach leads to a problem of code scattering and code tangling [21]. Scattering in agent-oriented models is the manifestation of design elements that belong to one specific concern, over several modeling units referred to other MAS concerns. Tangling in agent-oriented models is the mix of multiple concerns together in the same modeling elements. Using AOP, these problems can be efficiently solved since AOP gives the more flexibility to implement these concerns by integrating the aspects with the base code without changing the existing code.

AOP has various constructs such as: Joinpoint, Pointcut, Advice and Aspect [16].

- A joinpoint is a well-defined point in the code at which the concerns crosscut the application such as method call, object construction, or field access.
- A pointcut defines the times (joinpoint) in the application at which cross-cutting concern needs to be applied.
- An advice is a code fragment that is executed when join points satisfying its pointcut are reached. This execution can be done before, after, or around a specific join point.
- An aspect is a basic unit of modularization for crosscutting concern that cuts across multiple objects.

4 Proposed Aspect-oriented Trust Based Mobile Recommender System

In this section we present mobility crosscutting concerns in trust based multi-agent recommender system using agent-oriented approach with aspects. Section 4.1 presents the architecture of AOTMRS system. The functional description of our trust based multi-agent recommender system is discussed in section 4.2. Section 4.3 describes the consequences of not having explicit support for the modularization of mobility crosscutting concerns in the recommender system. Section 4.4 describes our approach to support these mobility crosscutting concerns in recommender system using agent oriented design enhanced with mobility aspect.

4.1 Architecture of the AOTMRS

AOTMRS automates the process of finding the appropriate places for the mobile user according to his demographic information. The basic building blocks of this system are shown in figure 1. On the client side, two components are active. The first is the normal off-the-shelf Internet browser and it is the only component that the user sees on the mobile screen during usual operation. Using this component only, the mobile user can interact with the recommendation system. The second component is the location detection that periodically sends the information about the mobile user such as his current location, time etc. to the mobility aspect at the server. This module is generally transparent to the mobile user. It will only display a startup dialog box for initialization purpose, for example to change the privacy settings or allowing the application to detect the mobile location.

Location dealer component of the server also contains two sub components: (i) mobility aspect and (ii) trust based recommendation system. The mobility aspect is in-charge of computing the location data i.e. transmitted by the location detection component of the client in order to obtain a good estimate of the user location (due to the power and computation capability limitations, it may be impractical for the mobile to compute the precise location, and only raw data are transmitted to the server) and to track the user's movements. The second sub component of location dealer, the trust based recommendation system is the core of the system: it maintains the access database and generates the recommendations for the user. Figure 1 shows the architecture diagram of AOTMRS.

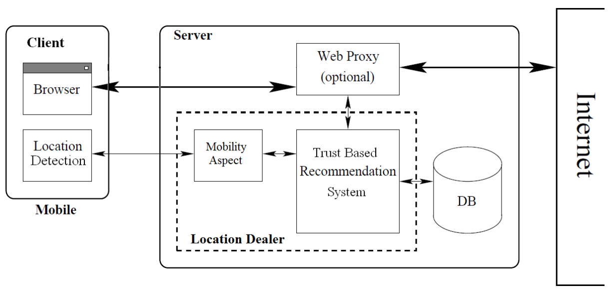

Figure 1: Architecture of the AOTMRS System

4.2 Function description of trust based multi-agent Recommender System

We developed trust based multi-agent recommender system for tourism that automates the process of finding the appropriate places for the user to visit according to his demographic information. Within this system the social recommendation process is taken into consideration by forming a network of the agents that act as a society and these agents interact with each other on the basis of trust relationships. These trustworthy relationships form a web of trust as illustrated in Figure 2, which shows such a network of peers represented by the numbered circles, where the numbers in the circles identify the various peers in the application domain. An edge represents that trustworthy relationship exists between the connected agents with certain degree of trust. Every agent in the system maintains a list of peers adjacent to it along with the degrees of trust on them. However, it is not necessary that if A trusts B with degree of trust as x, then B also trusts A with degree of trust as x.

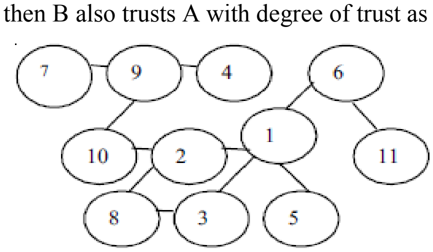

Figure 2: A network of peers represented by the numbered circles.

The peer agents in the community environment exchange recommendations about the products during their idle time, which is being referred to as unintentional encounters. This trust based multi-agent recommender system encompasses 3 types of software agents: (i) user agents (UA), (ii) recommender agents (RA) and (iii) information agent (IA).

The basic functionality of UAs is to infer and keep information about the corresponding mobile users who want the recommendations from the system. UA stores trust values for each interacting RA known to it and prioritize the RAs according to their trust value. UA collects the query for recommendations from mobile device of the user. UA passes this query to its trustworthy RAs only, for recommendations generation. This reduces the communication overload among the UA and RAs with the system as well as it also reduces no of computations perform by each RAs. Once, UA receives the recommendations from all trustworthy RAs in its neighborhood, it generates the final recommendation list of places after taking into account the degree of trust on each of the RA. UA also updates the degree of trust on each RA after evaluating the recommendations given by them.

The RAs represent those mobile users being referred as recommenders within the system, whose preferences are considered for generating the recommendations for the target mobile user. RA receives a query request from UA and then RA generates the recommendations for UA in the form of IFS. The IFS recommendation for a place to visit has a degree of membership (satisfaction), degree of non-membership (dissatisfaction) and the degree of hesitation (uncertainty) signifying the relevance, irrelevance, and uncertainty of the place for a given user. To personalize the place recommendations according to the taste of the user agent A, the RA maintains the following lists:

- *Preference list:* The preference list, P_A consists of the information in terms of attributes (renowned, distant, pleasurable, transport facility, climate, reasonably priced etc) of the places liked by the user in the past connected to user agent A. There are separate sub lists in P_A corresponding to the attributes renowned, distant, pleasurable, transport facility, climate, reasonably priced. The order of names of places in a sub list in P_A corresponding to a particular attribute signifies their priority in their respective sub lists.
- *Uncertain list:* The uncertain list, U_A consists of the same type of information as that of the preference list. However, it is unordered list accumulated during unintentional encounters and the RA has no idea whether the user prefers one place over the other.

IA manages the system information that is mainly stored in a database, and provides information to the other system agents as requested. IA stores the profiles of all registered mobile users with the system in his local database. IA also stores the details of all places along with their longitude/latitude values within the database. UAs and IA need to move in some circumstances, including when they are playing a specific role. For example, when the user first time asks for the recommendations then his corresponding UA has no trust on any RA. In this case UA needs to consult that RA profile, which is similar to the user's profile. In order to get that profile UA collaborates with an IA and requests this agent to search the information of the specified RA. IA controls the local database and is able to query for the profile. However, if the information is not available in the database, IA needs to move and try to find out the missing profile in remote environments. Although this mobility concern is separated from collaborative activities (roles), it is mixed with agent classes and methods that implement the agent's basic functionality. So if an agent has multiple roles and mobility actions, then the code of that agent will be more complex.

The functional diagram of trust based multi-agent recommender system is depicted in figure 3. Figure 4 presents mobility concern: crosscutting roles, plans and agent types in trust based multi-agent recommender system.

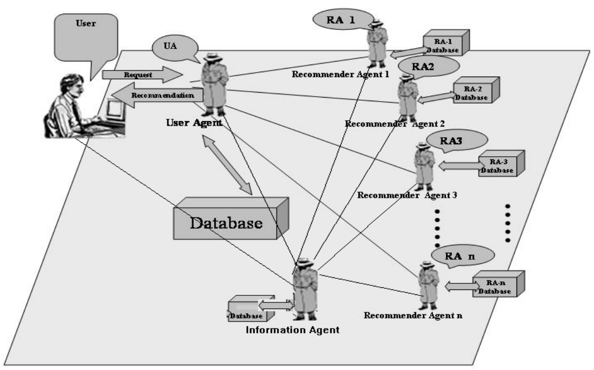

Figure 3: The Functional diagram of trust based multi-agent recommender system

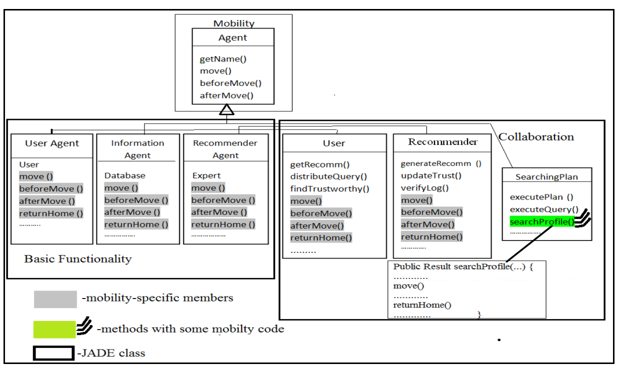

Figure 4: Mobility concern: crosscutting roles, plans and agent types

4.3 *Inherent problems to support mobility crosscuttings in design of Trust Based Multi-Agent Recommender System*

The fact that agents and its roles change their locations should not affect their basic functionality. However, the use of object-oriented abstractions does not provide support for the separation of the mobility concern and other agent concerns, such as the basic agent functionality and the agent's collaborative activities. The design and implementation of the mobility concern tend to affect or crosscuts many classes. As a consequence, it is difficult to understand the agent's basic functionalities, collaborations among agents, and mobility strategies of individual agents as a whole, which in turn decreases the system reusability and maintainability. So how do agents incorporate the mobility property without having their basic functionalities intertwined with the mobility concern is a critical problem. The various other inherent problems and their side effects related to crosscutting concerns, which cannot effectively handle in design of the multi-agent based recommender system such as necessary information missing, hindering of modular and compositional reasoning, replication of agent-oriented design elements, reduced evolvability and reuse opportunities are discussed in our prior work [5, 6].

4.4 *Enhanced Trust Based Multi-Agent Recommender System with Aspect Oriented Approach*

Mobility aspect is created and integrated with various agents' goals of trust based multi-agent recommender system at well defined join points with point cuts to improve the separation between the mobility concern and other agent concerns. The mobility aspect connects the execution points (events) on different agent classes in order to change their normal execution to handle mobility crosscutting concerns. The execution points include the execution of actions on plans, and agent types. Mobility aspect eliminates the problem of tangling and scattering from the trust based multi-agent recommender system easily because it provides better modular support for the separation of crosscutting concerns. If some mobility strategy changes later on, even then these can be easily handled by modifying this mobility aspect without changing the existing code of the recommender system. Hence our approach reduces the cognitive complexity and provides better modular way to design a multi-agent recommender system.

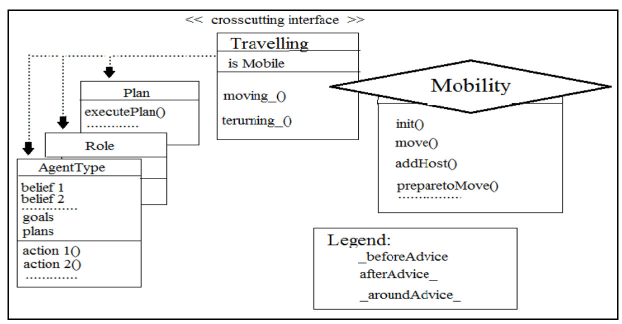

Figure 5: The static view of the Mobility aspect

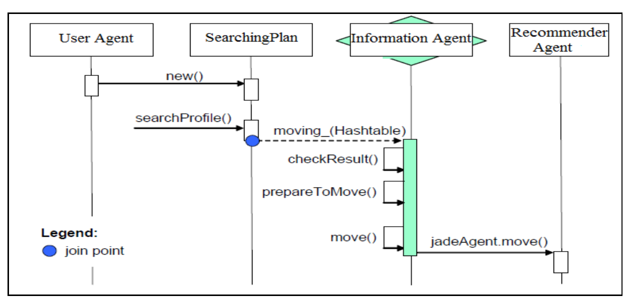

Figure 6: Dynamic behaviour of the Mobility aspect

Mobility aspect defines how and when the agents will move to other environments. The mobility aspect itself invokes the methods responsible for implementing the mobility-specific actions, such as identifying user position, move() and returnHome() etc. No mobility code remains in the agent classes that implement the basic functionality and collaborative activities. Figure 5 represents static view of the mobility aspect of trust based multi-agent recommender system. Figure 6 illustrates the dynamic behavior of the mobility aspect. The scenario shows that how the mobility aspect move agents and roles in a transparent way.

5 Trust Based Recommendation Generation

The recommendation process starts, once the UA receives the query request for recommendations from the mobile user. UA passes this query to its all trustworthy RAs. This query request contains explicit user preferences in terms of attribute rating value of renowned, distant, pleasurable, transport facility, climate, reasonably priced etc for the recommendation generation as well as implicit preferences such as user location in longitude/latitude format along with current time. Then RA filters out the places that are not located close to the current location of the mobile user from its own preference list (P_A) and uncertain list (U_A) by using following steps:

Step 1) RA matches all the places available in its P_A and U_A within its own local database for finding the corresponding longitude and latitude value for each place.

Step 2) If entries are found then go to step 4 else go to step 3.

Step 3) RA sends the remaining places list to IA for getting their corresponding longitude and latitude values. IA looks its database for those places and then sends corresponding longitude and latitude values back to the RA and then RA updates its local database.

Step 4) RA calculates the distance between each place of its P_A and U_A with the current location of mobile user by using the following formula:

$$Dist_j = r \times \arccos \begin{bmatrix} \sin(lat_i) \times \sin(lat_j) + \\ \cos(lat_i) \times \cos(lat_j) \times \cos(long_j - long_i) \end{bmatrix}$$

Where

$Dist_j$ is the distance between the j^{th} place and current user location

r is the radius of the earth (6378.7 Kilometers)

lat_i is the latitude of current user location in radians

$long_i$ is the longitude of current user location in radians

lat_j is the latitude of the j^{th} place of P_A or U_A in radians

$long_j$ is the longitude of the j^{th} place of P_A or U_A in radians

Step 5) RA prepares the temporary preference list (P_{TempA}) and temporary uncertain list (U_{TempA}) by filtering out the places belong to P_A and U_A, their $Dist_i$ value is less than given threshold value. This P_{TempA} and U_{TempA} list will be considered further for generating the recommendations.

5.1 *Generating Recommendations*

Various ways in which RA recommends a place are:

Case 1: The place is in the temporary preference list P_{TempA} corresponding to the user stored by RA.

Case 2: The RA comes to know about the place through a trustworthy acquaintance during unintentional encounters.

Case 3: The recommender has visited the place, and feels that the place has a general appeal even if it does not conform to the taste of the user. RA recommends that place to the user with the degree of uncertainty. The degree of membership is zero for such places and the degree of non-membership is computed using the other two degrees.

The places of Case 3 are recommended whether these are according to the taste of the user or not. For the places of Case 1 and Case 2, matching is done with P_{TempA} and U_{TempA}, respectively and based on the matching results, the recommendations are generated.

5.2 *Intuitionistic Fuzzy Set Generation for Places*

A recommender can recommend places known to it. The RA comes to know about the place either through usage or through unintentional encounters. During the unintentional encounters, an agent exchanges the information about only those places that it has experienced and is satisfied with.

An agent stores the names of places visited earlier. When the RA has to generate recommendations for other agents, it retrieves knowledge about the visited places.

Let a place P be represented by n attributes ($a_1, a_2 \ldots. a_n$). A place P is suggested to the user agent (UA), along with the IFS generated for it as shown below:

The degree of membership of a place P, μ_p is computed using the temporary preference list P_{TempA}, as:

For every attribute a_i (i = 1, 2..., n) of place P, do the following:

Search the position p_i of place P in temporary preference list P_{TempA} then compute the rank $Rank_i$ as the normalized position of P in the range [0 to 1] using the following formula:

$$Rank_i = 1 - ((p_i - 1)/(\max - 1))$$

where

max represents total number of places that exist in P_{TempA}

Finally, degree of membership of place P, μ_p is computed as:

$$\mu_P = \frac{\sum_{i=1}^{n} (da_i \times rank_i)}{n})$$

where

da_i (i = 1, 2, ., n) represents the normalized degree of significance that the user associates with the i^{th} attribute. n represents the total number of attributes of place P in P_{TempA}.

The degree of uncertainty of place P, π_p is computed using the temporary uncertainty list U_{TempA} as follow:

Let k_i be the attribute value of a place P for attribute a_i in U_{TempA}, then compute the degree of uncertainty of the place P as:

$$\pi_P = \frac{(da_1 \times k_1 + da_2 \times k_2 + \ldots + da_n \times k_n)}{n}$$

where

da_i (i = 1, 2, ., n) represents the degree of significance that the user associates with the i^{th} attribute.

n represents the number of places in U_{TempA}.

The degree of non-membership of a place P, ν_p is computed as follows:

$$\nu_P = 1 - \mu_P - \pi_P$$

5.3 *Recommendation List Generation by RA*

After matching the places with P_{TempA} and U_{TempA}, the degree of membership, degree of non-membership and uncertainty is available with the RA for all the places that it knows. The following method is used to generate the final list of the places that are to be recommended to the UA along with IFS that is computed for these:

1. All the places that are part of Case 3, section 5.1 are to be considered for further processing.

2. For all the places of Case 1 and Case 2, Section 5.1, do the following:

   2.1 The places with non-zero degree of membership are followed by the places with non-zero degree of uncertainty.

   2.2 Within the places with non-zero degree of membership, order the places in descending order on degree of membership.

   2.3 Within the places with non-zero degree of uncertainty, order the places in descending order on degree of uncertainty.

5.4 Final Trust-based Recommendation List Generation by User Agent (UA)

The UA needs to form an aggregated list out of the IFS recommendations lists received from the trustworthy RAs. UA has to generate a final consolidated list from all the recommendations that are received from the trustworthy RAs. The UA computes the degree of importance of a place on the basis of degree of trust on the RAs who have recommended the place, the relative position of the place in the list of RAs, and the IFS recommendation of the RA. The UA generates a final consolidated list from all the recommendations that are received from the recommenders using the following aggregation method:

1. Identify the distinct places from the lists and then compute the degree of importance (DoI) of every place (P_i) as follows:

$$DoI_i = MAX(ABS(DoT(R_1)\times\{\mu_i(R_1)-\nu_i(R_1)\times\pi(R_1)\}\times Rank_i(R_1)), ABS(DoT(R_2)\times\{\mu_i(R_2)-\nu_i(R_2)\times\pi_i(R_2)\}\times Rank_i(R_2)),\ldots\ldots\ldots ABS(DoT(R_k)\times\{\mu_i(R_k)-\nu_i(R_k)\times\pi_i(R_k)\}\times Rank_i(R_k)))$$

where

$DoI_i(A)$ is the degree of importance of P_i as computed by A

R_j is the j^{th} recommender

$\mu_i(X)$ is the degree of membership of P_i according to recommender X

$\nu_i(X)$ is the degree of non-membership of P_i according to recommender X

$\pi_i(X)$ is degree of uncertainty or hesitation of P_i according to recommender X

$DoT(R_j)$ is the degree of trust of A on R_j

$Rank_i(R_j)$ is the normalized position of P_i in the recommendation list of R_j

k is the total number of recommenders who have recommended P_i

2. Compute the threshold, TDOI for degree of importance as

$$\text{TDOI} = \mu - \nu - \pi$$

where,

μ, ν, and π are degree of membership, non-membership, and uncertainty, respectively that the UA expects from the interesting place.

3. For all the distinct places, P_i of Step 1, the degree of importance obtained is a real number that lies between 0 and 1; and hence, there is no need for its normalization. Arrange the products in the descending order of their degrees of importance.

5.5 Initializing and Updating Degree of Trust on the RAs

5.5.1 Trust Initialization

When a new agent comes up in the system or the system starts from the scratch, then the agents have to initialize the trust values for some of the other agents in the application domain to form its acquaintance set. If an agent is known to the other agent (i.e., the corresponding human know each other), then the human associated with the agent can initialize the degree of trust according to the personal dealings with the person. However, the system also allows an agent to initialize degree of trust on an agent X, on the basis of the experiences of the other agents with X, i.e., to what extent the other agents in the application domain have received good recommendations from X. The degree of trust is then regularly updated on the basis of the personal experience of the agent with X. The new agent Y, asks for the experience of known agents (connected to it) with respect to X. Let q agents return their experience values as the number of good recommendations received to the total number of the recommendations received from X. Let j^{th} agent gives the experience as e_j. Then the degree of trust on X as computed by Y can be written as follows:

$$DoT(X) = \frac{\sum_{j=1}^{q} e_j}{q}$$

If q is large, then basically one is interested in finding what the experience of majority of the agents is? Hence experiences in such cases can be clustered and degree of trust can be computed.

5.5.2 Trust Updation

The degree of trust on a recommender is updated on the basis of the distance between degree of importance of the place as given in the aggregated list of the user agent (A) and the recommendation list of the recommender (R). The difference of opinion between the user and the recommender is computed as follows:

$$d = \frac{(D_1 + D_2 + \ldots + D_P)}{p}$$

Where,
$D_i = \{ \mu_i(R) - \nu_i(R) * \pi_i(R) \} - \{ \mu - \nu * \pi \}$, μ, ν, π; and $\mu_i(R)$, $\nu_i(R)$, $\pi_i(R)$ are as defined earlier. P is the total number of places in the recommendation list of R. Depending upon whether the difference between its aggregated list and the recommendations is below its acceptable threshold d_t or not, the user agent updates the degree of trust, DoT(R) on recommender as follows:

$$DoT(R) = DoT(R) + (d_t - d)$$

Hence, trust increases for those recommenders who give good recommendations and vice versa.

6 Proposed Personalized Visual Interface of AOTMRS System

Users access a recommender system through its e-face, i.e., the interface. Visually attractive interface increases the trust of the user on the system and hence helps the system in retaining its users by ensuring their revisits to the system. We developed this personalized visual interface with the intention of achieving following goals:

- The user must be able to control the personal preferences used to create recommendations.
- The system must allow the user to express his opinion about the services presented.

The proposed visual interactive interface of AOTMRS reduces user interaction effort in order to provide more effective and persuasive recommendations. These recommendations are presented to the users keeping in view their personal requirements and corresponding product rating for every recommendation (given by the similar users). This visual interface of AOTMRS is interactive and customized according to user's preferences. User can amend his current requirements interactively in order to obtain the improved recommendations from the recommender system.

The AOTMRS requires explicit preferences in the form of grading of attributes of places for generating the recommendations. The system also implicitly calculates the grade for each of the place attributes, according to the user profile by using following step:

Let any place P is represented by n attributes (a_1, a_2, a_3…..a_n). For each attribute a_i it does the following steps.

- The system calculates the maximum frequency of place for each attribute value of a_i.
- The highest frequency attribute value will be selected as default grade.

The user may change the default grading given by the system through this interactive interface by clicking the up and down arrows as shown in figure 7. Once user clicks on "Enter" button the recommendations will be generated as markers, which indicate places within a map as shown in figure 8.

Figure 7: Snapshot represents grading of attributes for recommendation generation

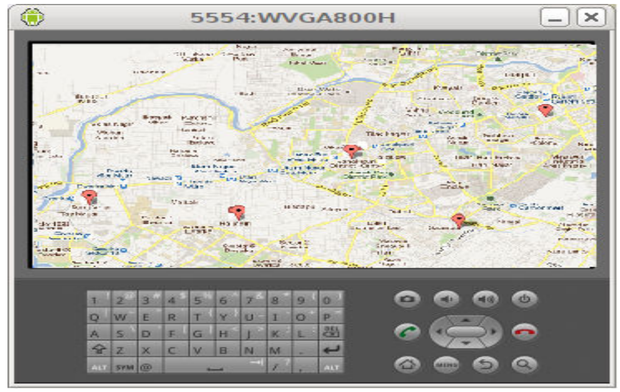

Figure 8: Snapshot represents recommendation of places

We have focused on making the process of giving feedback as simple as possible. The user can open the dialog of a place in the map after clicking on the markers, and hit the "rate"-button. This will present the user with the screen as shown in figure 9. The user can then apply his finger to select the preferred number of stars as shown in figure 10.

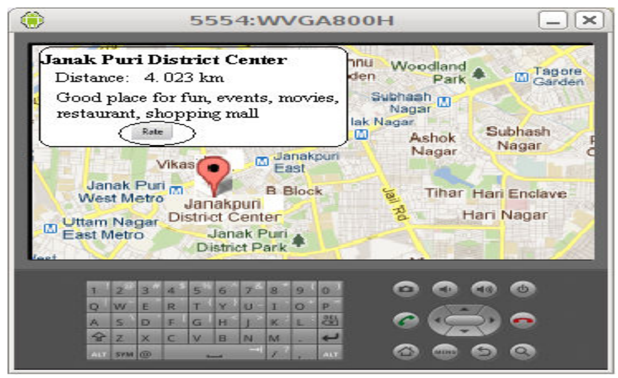

Figure 9: Snapshot represents details of recommendations

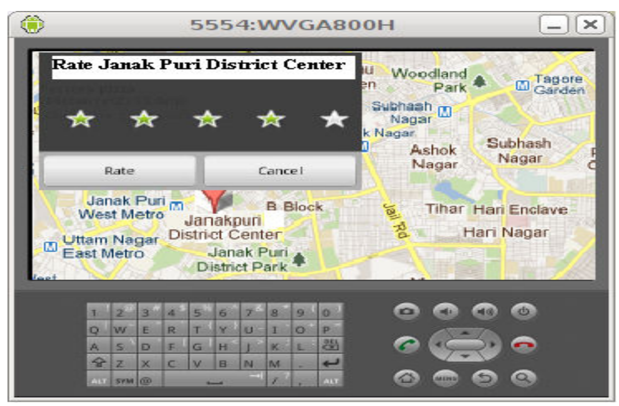

Figure 10: Snapshot represents user feedback in terms of ratings

7 Experimental Study

A prototype of the AOTMRS is designed and developed using JSP (Java Server Pages), Oracle 10g, JADE (Java Agent Development Environment), and AspectJ on Eclipse interface. The location detection component of AOTMRS is implemented by using the Geolocation API of HTML 5. We collected the information of various tourist zones, tourist states, tourist cities, hill stations, tourist places and available hotels at these places in India from www.touristplacesinindia.com website. This collected dataset contains the details of 4 tourist zones, 11 tourist states, 27 tourist cities, 7 hill stations, 159 tourist places and 437 available hotels at these places. We calculated the longitude and latitude values for each collected tourist places by performing reverse geo-coding using google map. This experimental setup consisted of 15 mobile users who are the research scholars for result evaluation.

AOTMRS system creates an agent corresponding to each mobile user, when they register with the system. The user agent has a certain degree of trust on the recommender agents, which is represented in the Table 1. In Table 2, degrees of significance that the user agent associates with the attributes of a place are shown. User agent receives the query regarding place recommendation from the mobile user and passes this query to its trustworthy recommender agents. The recommender agents respond with the degrees of membership and non membership about the places, as shown in Table 3. Finally Table 4 shows the aggregated list of all the recommendations as computed by the user agent and given to the target mobile user. The degree of importance is negative for those places that do not conform to the taste of the user but have been suggested as these have mass appeal.

Recommenders	**R_1**	**R_2**	**R_3**	**R_4**	**R_5**
Degree of Trust	0.76	0.85	0.33	0.79	0.68

Table 1: The degree of trust on RAs according to the UAs

Renowned	Distan t	Climat e	Pleasurable	Transport Facility	Economica lly Priced
0.5763	0.7623	0.2345	0.3323	0.3728	0.2224

Table 2: Degree of significance of attributes of place

		R_1	**R_2**	**R_3**	**R_4**	**R_5**
Embience Mall	μ	0	0.9125			
	ν	0.1732	0.8323			
District Center	μ		0			0.7923
	ν		0.2332			0.2099
Rajori Garden	μ	0.2356		0.1246		
	ν	0.7685		0.8745		
District Park	μ		0.8339		0.9214	
	ν		0.1667		0.07823	
Fun and Food Villege	μ		0.1179		0.0058	0.01443
	ν		0.8821		0.9942	0.8957
Adventure Ireland	μ	0		0		0
	ν	0.89		0.63		0.64
PVR Vikar Puri	μ	0.7768		0		0.9072
	ν	0.2345		0.2		0.0927

Table 3: Recommendation of five RAs in the form of IFS

Places	**Degree of Importance**
District Center	0.6347
Embience Mall	0,5136
Adventure Ireland	0.5095
Fun and Food Village	0.3995
PVR Vikar Puri	0.2309
Rajori Garden	0.08967
Adventure Ireland	-0.07854

Table 4: The aggregated list as obtained by UA

To be successful, a recommender system must be able to effectively guide a user through a product-space and, in general, higher precision is to be preferred. For this evaluation, we find the number of places viewed by users before they accepted the system's recommendation (precision) compared against the recall values (ratio of the relevant places recommended by the system and places preferred by the user in the actual recommendation list). Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or quantity.

Precision is obtained here as a ratio between the number of places both relevant and retrieved in the recommendation list and the number of places retrieved by the recommender system for a particular user query. Recall is obtained as a ratio between the number of places both relevant and retrieved in the recommendation list and the number of places actually relevant. We compared the AOTMRS system's precision and recall value with their values of collaborative filtering based recommender system.

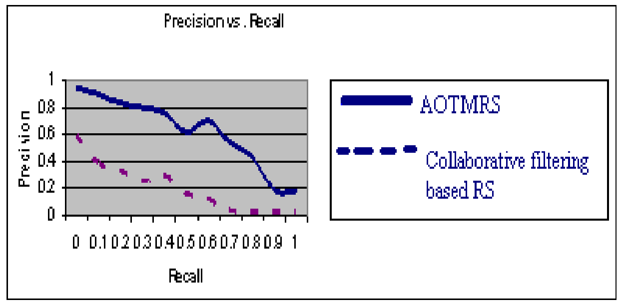

Figure 11: Precision vs. Recall Graph

We also evaluated all SoC (Separation of Concern) measures such as CDC (Concern diffusions over components), CDO (Concern diffusions over operations) and CDLOC (Concern diffusions over LOC) and coupling metric such as CBC (Coupling between components) [10] for trust based multi-agent recommender system with and without using aspect. The CDC metric detected that recommender system required more components for implementing mobility concern in recommender systems using conventional agent-oriented approach than the AOP approach. CDO metric required more operations (methods/advices) in the conventional agent-oriented solution than in the AOP solution. The CDLOC metric also pointed out that the AOP solution was more effective in terms of modularizing the mobility concerns across the lines of code.

Table 5 presents the comparative results of all SoC and coupling measures for user mobility in the trust based multi agent recommender system with and without using aspects.

Concerns	**Separation of Concerns**			**Couplin g**
	CDC	**CDO**	**CDLOC**	**CBC**
Mobility Concern without aspect	4	9	5	3
Mobility Concern with aspect	2	5	2	1

Table 5: Results of SoC measures

8 Conclusion and Future Work

Aspect-Oriented Trust Based Recommender System (AOTMRS) is presented in this paper to handle user mobility aspect of multi-agent based recommender system in a better modular way and reducing the cognitive complexity. The main emphasis in the presented work is the identification of mobility crosscutting concern and embedding of this concern with the system by creating mobility aspect to improve the separation of mobility concern in recommender system. We designed and developed the prototype system AOTMRS for tourism recommendations. The problems of tangling and scattering is solved in the presented work for handling mobility crosscutting concerns in recommender system with aspect using AOP. The mobility concern is totally modularized in AOTMRS, so the basic agent classes requires no mobility behavior and do not need to be changed as changes in the mobility strategy is required. Since mobility strategies in a multi-agent based recommender system may vary as the system evolves so AOTMRS provides more flexible design to support these changes in multi-agent based recommender system.

The social recommendation process is integrated within AOTMRS in the form of IFS having membership, non-membership and hesitation part. The concept of trust has been incorporated in this paper to match the human thinking process. Trust is also updated by the user on recommender based on the recommendation received. In the proposed system, we have tried to merge the advantages of the mechanical recommender system with the more human recommendation process to make their recommendations trustworthy and useful for the mobile user. The personalized visual interface is also developed for reducing the user interaction effort to overcome the limitation of mobile devices.

As a future work, we will be working towards a proactive mobile recommender system. A proactive recommender system pushes recommendations to the user when the current situation seems appropriate, without explicit user request

References

[1] K. Atanassov, “Intuitionistic Fuzzy Sets: Theory and Applications”, Studies in Fuzziness and Soft Computing, Vol. 35, Physica-Verlag, 1999.

[2] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of the-art and possible extensions”, IEEE Trans. Knowledge Data Engg., 17(6), pp 734-49, (2005).

[3] Amir, Albadvi and & M. Shahbazi, “A hybrid recommendation technique based on product category attributes”, Int. J. Expert Syst. Appli., 36(9), 11480-488, (2009).

[4] P. Bedi, A. Sinha, S. Agarwal, A. Awasthi, G. Prasad and D. Saini, “Influence of Terrain on Modern Tactical Combat: Trust-based Recommender System”, Defense Science Journal, India, Vol. 60, No. 4, pp. 405-411, (2010).

[5] P. Bedi, and S. Agarwal, “Managing Security in Aspect-Oriented Recommender System”, International Conference on Communication Systems and Network Technologies, pp 709-713, Katra, India, June (2011), IEEE Computer Society.

[6] P. Bedi, and S. Agarwal, “Preference Learning in Aspect-Oriented Recommender System”, International Conference on Computational Intelligence and Communication Networks, pp 611-615, Gwalior, India, October (2011), USA: IEEE Xplore.

[7] P. Bedi, and S. Agarwal, “Aspect-Oriented Mobility-Aware Recommender System”, International Conference on World Congress on Information and Communication Technologies, pp 191-196, WCIT 2011, December 11-14, 2011 at Mumbai, India, USA: IEEE Xplore.

[8] P. Bedi, H. Kaur and S. Marwaha, “Trust based recommender system for the semantic web”, 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, (2007).

[9] S. Burigat, L. Chittaro and L. Marco, “Bringing dynamic queries to mobile devices: A visual preference-based search tool for tourist decision support”, volume 3585 of Lecture Notes in Computer Science, pages 213–226, Springer, (2005).

[10] N. Cacho, C. Anna, E. Figueiredo, A. Garcia, T. Batista, and C. Lucena: “Composing design patterns: a scalability study of aspect-oriented programming”, AOSD, pp 109-121, (2006).

[11] K. Church and B. Smyth, “Who, what, where & when: a new approach to mobile search”, Intelligent User Interfaces, pages 309–312, ACM (2008).

[12] M. Dunlop, A. Morrison, S. McCallum, P. Ptaskinski, C. Risbey and F. Stewart, “Focussed palmtop information access through star field displays and profile matching”, In Proceedings of the Workshop on Mobile and Ubiquitous Information Access, pages 79–89, Glasgow, Scotland (2004).

[13] N. Good, B. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, “Combining collaborative filtering with personal agents for better recommendations”, In Proceedings of Conference of the American Association of Artificial Intelligence AAAI-99, July (1999).

[14] K. Honda, A. Notsu and H. Ichihashi, “Collaborative Filtering by User-Item Clustering Based on Structural Balancing Approach”, IJCSNS International Journal of Computer Science and Network Security, Vol 8, December (2008).

[15] G. Hussein, “Enhanced K-means-Based Mobile Recommender System”, International Journal of Information Studies, Volume 2 Issue 2, April (2010).

[16] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C., Lopes, J. Loingtier and J. Irwin, “Aspect-Oriented Programming”, ECOOP, 220-242, (1997).

[17] T. Mehmood, N. Ashraf, K. Rasheed and S. Rehman, “Framework for Modeling Performance in Multi Agent Systems (MAS) using Aspect Oriented Programming (AOP)”, The Sixth Australasian Workshop on Software and System Architectures (AWSA 2005), pp 40-45, Brisbane Australia, (2005).

[18] M. Park, J. Hong and S. Cho, ”Location-based recommendation system using bayesian user’s preference model in mobile devices”, volume 4611 of Lecture Notes in Computer Science, pages 1130–1139, Springer (2007).

[19] F. Ricci and Q. Nguyen, “Acquiring and revising preferences in a critique-based mobile recommender system”, IEEE Intelligent Systems, 22(3):22–29 (2007).

[20] P. Tangphoklang, C. Tanchotsrinon, S. Maneeroj and P. Sophatsathit, "A Design of Multi-Crieteria Recommender System Architecture for Mobile Banking business in Thailand", In Proceedings of the Second Internation Conference on Knowledge and Smart Technologies, pp 45-48, July 24-25(2010).
[21] P. Tarr, H. Ossher, W. Harrison and S. Sutton, "N Degrees of Seperation: Multi Dimentional Seperation of Concerns", 21st International Conference on Software Engineering, pp 107-119, (1999).
[22] G. Tumas and F. Ricci, "Personalized mobile city transport advisory system". Information and Communication Technologies in Tourism, pages 173–184. Springer, (2009)
[23] F. Yang, Z. Wang, "A Mobile Location-based Information Recommendation System based on GPS and WEB2.0 Services", WSEAS Transactions on Computers, pp 725-734, issue 4, vol 8, April 2009.
[24] K. F. Yeung, Y. Yang, "A Proactive Personalized Mobile News Recommendation System", In Developments in E-systems Engineering, pp 207-212, IEEE Computer Society (2010)

Author Biographies

Dr. Punam Bedi received her Ph.D. in Computer Science from the Department of Computer Science, University of Delhi, India in 1999 and her M.Tech. in Computer Science from IIT Delhi, India in 1986. She is an Associate Professor in the Department of Computer Science, University of Delhi. She has about 25 years of teaching and research experience and has published more than 140 research papers in National/International Journals/Conferences. Dr. Bedi is a member of AAAI, ACM, senior member of IEEE, and life member of Computer Society of India.
Her research interests include Web Intelligence, Soft Computing, Semantic Web, Multi-agent Systems, Intelligent Information Systems, Intelligent Software Engineering, Software Security, Intelligent User Interfaces, Requirement Engineering, Human Computer Interaction (HCI), Trust, Information Retrieval, Personalization, Steganography and Steganalysis.

Mr. Sumit Kr. Agarwal received his M.Tech degree in Information Technology from USIT, Guru Gobind Singh Indraprasth University Delhi, India in 2007 and his M.Sc in Information Science from Dr. Bhim Rao Ambedkar University, Agra in 2002. He is a doctorate student in Department of Computer Science, University of Delhi. Mr. Agarwal is an associate member of IETE, Loadhi Road, New Delhi. His research interests include Web Intelligence, Multi-Agent Systems, Trust, Recommender Systems and Aspect-oriented Programming.