Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 8 (2013) pp. 177-188

Integrating Trust in Argumentation Based Recommender Systems

Punam Bedi[1], Pooja Vashisth[2*]

[1]Department of Computer Science,
University of Delhi,
Delhi, INDIA
punambedi@ieee.org

[2]Department of Computer Science,
University of Delhi,
Delhi, INDIA
poojavashisth@rediffmail.com

***Abstract*: With the enormous growth of the Internet, trust has become an increasingly important issue in Agent-based E-commerce. Argumentation technologies are needed for autonomous agents to come to mutually acceptable agreements, on behalf of humans. Agents can argue over each other's beliefs, desires and planning. It is also important for these agents to be able to compute their trust in other agents. Especially, in an argumentation-based recommendation system, the arguments uttered to persuade a customer over a product are not the result of an isolated analysis, but of an integral view of the preferences, goals and options available. In our opinion, trust and argumentation together can improve the recommendation process. This paper describes our work on using argumentation to handle and update trust in the agent-based recommender systems and vice versa. This paper proposes integration of fuzzy trust with an argumentation framework to enable the agents in reasoning about the beliefs, desires and plans with trust. This integration allows the user to take well-reasoned decisions based on trustworthy recommendations. As a result, trust in an agent increases when it generates more of acceptable arguments thereby reducing the number of messages passed and time consumed by the agents. This improves performance of the agents in terms of the communication overhead caused and time taken for decision making. The same was established by the results obtained from experiments conducted for a Book Recommender System (RS).**

***Keywords*:** Intelligent Agents, Trust, Argumentation Framework, Fuzzy logic, Planning.

I. Introduction

Recommendation Systems (RSs) are aimed at helping users to deal with the problem of information overload by facilitating access to relevant items suitable to their preferences. They attempt to generate a model of the user or user's task and apply diverse course of actions to anticipate what information may be of interest to the user [1], [2]. RSs have also been built based on social factors related to user's personality and trust to improve the satisfaction of users involved in the process [3], [4]. Although the effectiveness of existing recommenders is remarkable, they still have some serious limitations. Several recommendation systems lack the persuasive power required to convince the users [5].

In fact, quantitative approaches as opposed to qualitative approaches have often been criticized for their inability to obtain conclusions supported by a rationally justified procedure [6], [7]. The quantitative techniques adopted by most existing user support systems suffer also from this limitation. As a result, serious trustworthiness issues may arise, especially in those cases when business interests are involved, or when external manipulation is possible [8], [9]. Logic based approaches could help to overcome these issues, enhancing recommendation technology by providing a means to formally express constrains and to draw inferences [7]. Much of the work on trust in computer science has concentrated on dealing with specific scenarios in which trust has to be established or handled in some fashion. The internet, as the largest distributed system of all, is naturally a target of much of the research on trust. There have been studies on the development of trust in ecommerce through the use of reputation systems and studies on how such systems perform [10], [11]. Another area of concern has to do with the reliability of sources of information on the web, like the one provided by the recommender systems. In the work [12], the authors aim to support decision making in situations where the source of the information on which decisions are based is of varying trustworthiness. Their approach uses formal argumentation to capture the relationships between such information sources and conclusions drawn from them. The other works for example [13], [14], [15], investigates mechanisms to determine which sources to trust when faced with multiple conflicting sources and looks at the related question of how to resolve such conflicting information. In [14], the authors introduced a formal system of argumentation for reasoning using information about trust. Whereas the work in [15] follows a simple approach to reasoning about trust with logic, and describes how it can be combined with reasoning about beliefs using logic for decision making purposes. The work presented in [16] extends this idea to rate the individuals who provide information by looking at the history of the arguments they have provided. In a recent work [17], Bedi et al. proposed a hybrid RS that builds a trust model based on all accepted and unaccepted arguments generated by the agents in the system called Trust enabled Argumentation Based Recommender System (TABRS). This system uses Belief-Desire-Intention (BDI) agents enabled with reasoning and argumentation capability. These agents can argue about each other's beliefs, goals and plans using argumentation to generate trustworthy recommendations. Trust is an especially important issue from the perspective of autonomous agents and multi-agent systems as well. The premise behind the multi-agent systems field is that of developing software agents that will work in the interests of their owners, carrying out their owners' wishes while interacting with other entities [18]. In such interactions, agents will have to reason about how much they should trust those other entities, whether they are trusting those entities to carry out some task, or whether they trust those entities to not misuse crucial information. Therefore, trust plays a significant role in the agent-based systems.

Recommender Systems can provide personalized information services in varied ways. Therefore, over the years several inter-disciplinary techniques (like trust, soft computing techniques, etc.) have been applied to them to improve their efficiency. Soft Computing (SC) seems to be the appropriate paradigm to handle the uncertainty and fuzziness of the information available to model user's requirements and trust in the system [19], [20]. Amongst various SC techniques, the fuzzy logic field has grown considerably in a number of applications across a wide variety of domains for product recommendations [20], [21], [22]. The fuzzy logic theory has been the subject of interest to researchers in the recommender systems' field applied to information retrieval on web and e-commerce [23], because of its proven efficiency for solving problems of fuzzy nature in all areas. This theory is used to solve problems of these systems, giving a sort of intelligence to the system in some cases like the system presented in [24].

The paper is an amalgamation of the fuzzy logic techniques [25] and the argumentation based trust modeling as presented in the proposed RS [17]. The paper is organized as follows: section 2 briefly describes the argumentation framework of the recommender system (RS is based on BDI agents) used for integrating fuzzy trust with argumentation. Section 3 gives an overview of the basic fuzzy trust model used for the RS whereas section 4 describes how this basic trust model can be further improved by using argumentation (some work related to this was done earlier [26]). This section also covers the aspect of using argumentation for trust and vice versa. It shows how trust can be used to affect the strength of the instrumental arguments responsible for selection of a plan for execution. This way trust can be used to improve the argumentation for planning between agents. Section 5 deals with the integration of trust with the argumentation plan generation process for implementation purpose. Finally, section 6 presents the experimental evaluation of the above concepts which is then followed by discussions and conclusion.

II. Framework of an Agent Based Recommender System using Argumentation

In this section, we briefly present our argumentation-based framework for recommender systems [27].

Recently, argumentation has been gaining attention in the multi-agent community. Autonomous and social agents need to deliberate under complex preference policies, related to the environment in which they evolve. Argumentation can be defined as –"A social and verbal means of trying to resolve or at least to contend with, a conflict or difference that has arisen or exists between two (or more) parties". Interactions bring new information to the agents. Interaction using argumentation is an established approach for reasoning with inconsistent knowledge and hence helps in decision making [28], [29]. It can also be used for processing users' opinion and resolving various conflicts between them.

Agents in the proposed recommender system (RS) have a BDI architecture (Beliefs, Desires, and Intention) augmented with argumentation and logical reasoning capability. The agent architecture is composed of two models: the mental model, and the reasoning model. The mental model includes beliefs, desires, goals and plans. The agents must use their reasoning capabilities to reason about their mental states before taking any decisions. The agent's reasoning capabilities are represented by the reasoning model using an argumentation system. To deal with the different nature of the arguments involved, we have developed three distinct argumentation frameworks: *one for reasoning about beliefs, another for arguing about what desires should be pursued, and a third for arguing about the best plan to intend in order to achieve these desires.* These beliefs, desires and the related, supporting arguments can be used to generate an interesting recommendation or even to defeat one. During recommendation seeking process, agents can establish a common knowledge of each other's likes (satisfaction) and dislikes (dissatisfaction), find compromises, and persuade to make decisions through argumentation.

An argumentation framework (see figure 1) is simply a set of arguments and a binary relation representing the attack-relation between the arguments. The following definition, describe formally an argument. Here $\mathcal{K}_{\mathcal{B}}$ indicates a possibly inconsistent knowledge base. $\vdash$ stands for classical inference and $\equiv$ for logical equivalence. $\mathcal{D}$ denotes the set of desires, a base B_b contains agent's basic beliefs and $\mathcal{RES}$ denotes a set of resources available in the system.

Definition 1 (Argument). *An argument is a pair* (H, h) *where h is a formula of a logical language and H a sub-set of* $\mathrm{K_B}$ *such that i) H is consistent, ii)* $H \vdash h$ *and iii) H is minimal, so* *no subset of H satisfying both i) and ii) exists. H is called the support of the argument and h its conclusion.*

Definition 2 (Attack Relation). *Let (H_1, h_1), (H_2, h_2) be two arguments. (H_1, h_1) attacks (H_2, h_2) iff $h_1 \equiv \sim h_2$.*

Definition 3 (Basic beliefs of an agent). An agent's basic beliefs is a set $B_b = \{(\beta_i, a_i, b_i);\ i = 1, \ldots, n\}$, where β_i is a consistent propositional formula , a_i its degree of certainty and b_i its preference as per the agent. The degree of certainty and preference is required in order to generate an ordering over arguments, which is required by the underlying argumentation theory.

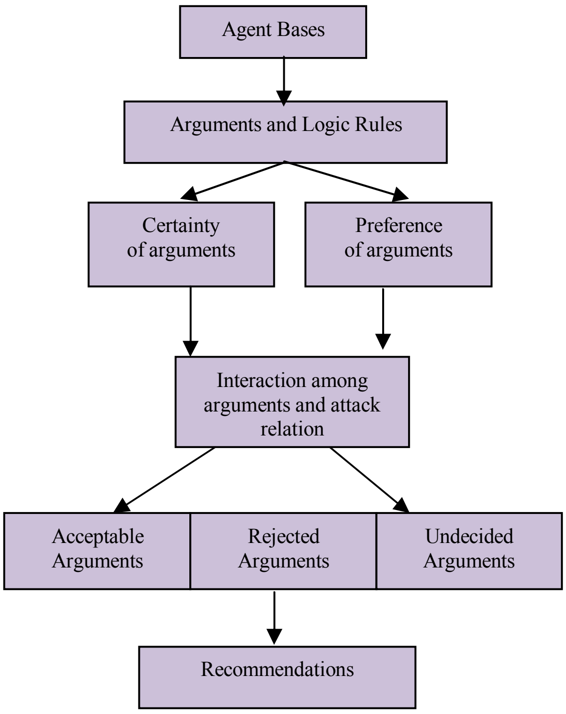

Figure 1. Conceptual view of an argumentation framework for the Recommender System

Before making a suggestion h, the speaker agent must use its argumentation system to build an argument (H, h). The idea is to be able to persuade the addressee agent about h, if he decides to refuse the suggestion. On the other side, the addressee agent must use his own argumentation system to select the answer he will give. To be able to communicate and argue, the agents use a set of logic rules based on the facts (beliefs, desires, intentions) stored in its $\mathcal{K}_{\mathcal{B}}$. From the knowledge base $\mathcal{K}_{\mathcal{B}}$, two kinds of rules can be defined: *desire-generation* rules and *planning* rules [30].

Definition 4 (Desire-Generation Rules or DGR). A desire-generation rule (or *dgr*) is an expression of the form:

$\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \ldots \wedge \psi_1 \wedge \ldots \wedge \psi_m \Rightarrow \psi$ *where* $\forall\ \varphi_i \in \mathcal{K}$ *and* $\forall\ \psi_i, \psi \in \mathcal{D}$.

The meaning of the rule is "if the agent *believes* $\varphi_1, \ldots, \varphi_n$ and *desires* $\psi_1, \ldots, \psi_m$, then the agent will *desire* ψ as well".

Definition 5 (Planning rules). A planning rule is an expression of the form:

$\varphi_1 \wedge \ldots \wedge \varphi_n \wedge r_1 \wedge \ldots \wedge r_m \rightarrowtail \varphi$ *where* $\forall\ \varphi_i \in \mathcal{D}$, $\varphi \in \mathcal{D}$ *and* $\forall\ r_i \in \mathcal{RES}$.

A planning rule expresses that if $\varphi_1, \ldots, \varphi_n$ are achieved and the resources $r_1, \ldots, r_m$ are used then φ is achieved.

Now, as per our argumentation framework, each agent is equipped with four bases: a base B_b containing its *basic beliefs*, a base B_d containing its desire-generation rules, a base B_p containing its planning rules and finally a base R which will gather all the resources possessed by that agent. Beliefs can be uncertain, desires may not have equal priority and resources may have different costs.

Definition 6 (Agent's bases). *An agent is equipped with four bases $<B_b, B_d, B_p, R>$:*

- $B_b = \{(\beta_i, a_i, b_i) : \beta_i \in \mathcal{K}_{\mathcal{B}}, a_i, b_i \in [0, 1], i = 1, \ldots, n\}$. *Triplet (β_i, a_i, b_i) means belief β_i is certain at least to degree a_i and β_i is preferred up to degree b_i by an agent.*
- $B_d = \{(dgr_i, s_i, p_i)$: dgr_i *is a desire generation rule,* $s_i \in \mathbf{R}$, $i = 1, \ldots, m\}$. *Symbol s_i denotes the strength of the desire ψ generated by the rule dgr_i and p_i denotes the preference for that desire. Let Strength$(\psi) = s_i$ and Preference$(\psi) = p_i$.* In the proposed framework the worth i.e. *Worth*(ψ) of a preferred desire depends on both, the certainty and preference associated with its antecedents. Therefore,

$$Worth(\psi) = \begin{cases} Strength(\psi) & ;\text{ if } p_i = 0 \\ Preference(\psi) & ;\text{ otherwise,} \end{cases}$$

where $Preference(\psi) = \dfrac{1}{n+m}\left(\sum_{i=1}^{n} a_i * b_i + \sum_{j=1}^{m} s_j * p_j\right)$.

- $B_p = \{pr_i : pr_i$ *is a planning rule*$\}$.
- $R = \{r_i,\ i = 1, \ldots, n\}$ *where* $r_i \in \mathcal{RES}$. *These resources appear in the plan and represent the material required to be consumed for satisfying a related desire.*

Later on, we describe influence and integration of trust in the argumentation framework for planning in section 4.

III. A Basic Fuzzy Trust Model for the Recommender System

Our fuzzy trust model for an argument-based recommender system is built on the following definitions.

Definition 7 (Trust). *Trust is a subjective expectation a partner has about another's future behavior based on the history of their encounters.*

These encounters in an argument-based recommendation system consist of acceptable, unacceptable arguments, satisfactory and unsatisfactory recommendations. These facts are computed from the agent's local beliefs.

Trust based systems are rating systems where each individual is asked to give his opinion after completion of each interaction in the form of ratings (it can be implicit or explicit). In our work, trust values are automatically inferred from the rating database of the RS and thereafter these values are used to enhance the accuracy of the recommendation process. These interactions consist of the recommendations by the RS. More formally, let $A = \{a_1, a_2, \ldots, a_M\}$ be the set of all agents (user as well as recommender agents), where M is the number of agents in the system. We assume each user agent will rate a recommender agent after completing the recommendation process. An interaction (recommendation) i $\in$ I, where $r_{xy}(i_k)$ is the rating agent x has given to agent y for an interaction (recommendation) i_k. The rating scale or grade for recommendations is defined as G = {-2, -1, 0, +1, +2}. The set of ratings agent x has given to agent y is $S_{xy} = \{r_{xy}(i_k) \mid i_k \in I\}$ and the whole past history of agent x is $H_x = \{S_{xy} \mid \forall\, y(\neq x) \in A\}$. The agent's rating for a recommendation i_k is derived implicitly from the number of matches the argument parameters strike with the agent's preference list. The rating can also be given explicitly by the user on scale 'G' as mentioned above.

Given that A is the set of agents. We define an agent's trustworthiness as follows: $TRUST : A \times A \times F \rightarrow [0,1]$. This function associates to each recommender agent a fuzzy measure representing its trustworthiness according to other user agents. To evaluate the trustworthiness of an agent y, an agent x uses the history of its interactions with y. Equation 1 shows how to calculate this fuzzy measure of trustworthiness for recommendations.

$$FM(R_{xy}) = \frac{\sum_{R \in I} FM(R_{xy}) * w_{satisfied} - \sum_{R \in I} FM(R_{xy}) * w_{unsatisfied}}{T\_N\_R_{xy}} \tag{1}$$

Where $FM(R_{xy})$ denotes the trustworthiness of y according to x's point of view.

$\sum_{R \in I} FM(R_{xy}) * w_{satisfied}$ is the summation of the fuzzy measure of the degree of x's satisfaction over y's recommendations obtained from eq. (4). Hence, these recommendations are acceptable to x.

$\sum_{R \in I} FM(R_{xy}) * w_{unsatisfied}$ is the summation of the fuzzy measure of the degree of x's dissatisfaction over y's recommendations obtained from eq. (5). Hence, these recommendations are unacceptable to x. Here, $w_{satisfied}$ and $w_{unsatisfied}$ are the weights attached to the acceptable and unacceptable recommendations respectively. These weights are determined directly from the strength of the instrumental arguments (see equation (14)) that triggered selection of the plans behind a given recommendation (explained in section 4.2.1). $T\_N\_R_{xy}$ is the total number of recommendations made by agent 'y' for agent 'x'; a count maintained by the agent's persistent belief base in the system. We now need to determine the user satisfaction (satisfied to what extent) and dissatisfaction (unsatisfied to what extent) for a recommendation generated by a recommender agent. To do so, we can define two fuzzy subsets on each agent's ratings, say satisfied and unsatisfied. This is because the fuzzy sets can clearly capture the concept of finding the extent (membership value) to which an agent is satisfied or unsatisfied with an interaction or recommendation. The satisfied and unsatisfied fuzzy subsets for agent x are defined as below:

$$\text{satisfied}(x) = \{sat_x(i_k) \mid i_k \in H_x\} \tag{2}$$

$$\text{unsatisfied}(x) = \{unsat_x(i_k) \mid i_k \in H_x\} \tag{3}$$

where $sat_x(i_k)$ and $unsat_x(i_k)$ are membership values of x's ratings for a recommendation i_k in the fuzzy subsets satisfied(x) and unsatisfied(x), respectively.

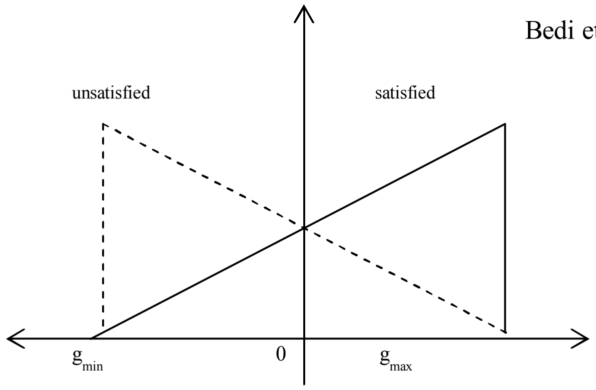

Figure 2. Fuzzy membership function to find the degree of satisfaction and dissatisfaction for an interaction

Now, we give a simple triangular membership function (see figure 2) for satisfied(x) and unsatisfied(x) fuzzy subsets that are defined by the following popular equations [25]:

$$sat_x(i_k) = \begin{cases} 0 & r_{xy}(i_k) = g_{min} \\ \dfrac{r_{xy}(i_k) - g_{min}}{g_{max} - g_{min}} & g_{min} < r_{xy}(i_k) < g_{max} \\ 1 & r_{xy}(i_k) = g_{max} \end{cases} \tag{4}$$

$$unsat_x(i_k) = 1 - sat_x(i_k) \tag{5}$$

where g_{min} and g_{max} are the minimum and the maximum ratings for a given system with G = { g_{min}, …, 0, …, g_{max}}.

IV. Trust and Argumentation

Argumentation can be seen as the principled interaction of different, potentially conflicting arguments, for the sake of arriving at a consistent conclusion. Argumentation can give us means for allowing an agent to reconcile conflicting information within itself, its informational state and between multiple agents through communication. It can also be used to justify an observed behavior, and therefore modify the impact on the model of expected behavior that is at the base of trust models [31], [32], [33]. There are (at least) two forms of integration, beneficial to both models:

- An argumentation process can be used to improve the performance of a trust model (i.e. argumentation for trust).
- A trust model can be used to determine the reliability of an argument (i.e. trust for argumentation).

Before considering the above mentioned aspects of the integration in detail, we now brief the interplay between trust and argumentation from the (proposed) system's perspective. Trust can get affected by the interaction between the agents. An agent can accept or reject any information from another agent or take a decision in support or against. Trust is updated by both direct and indirect interactions. For updating trust we considered users' satisfaction and dissatisfaction with the interactions and recommendations. Trust can change with changes in the agents' beliefs. It can also get affected due to argumentation. An argument can be formed due to an agent's own beliefs or it can also be based on the collective information from the agent's trustworthy counterparts. As a result, whenever an argument attacks or supports, this may result in the winning situation for the argument under consideration. Such outcomes lead to changes in trust. This is because; if the current user likes this winning argument then the trust in all the users who have participated in providing information for the argument increases by a certain amount. A user dislike or a defeated argument may as well lead to decrement in trust.

A. Argumentation for Trust

The argumentation approach can be used to improve the performance of the trust mechanisms. This can happen when trust is computed by an agent in isolation (relying on past interactions with a target) or when agents can exchange and share information about the trustworthiness of possible targets (or one another). Computing trust is a problem of reasoning under uncertainty, requiring the prediction and anticipation by an agent (the evaluator) of the future behavior of another agent (the target). Despite the acknowledged ability of argumentation to support reasoning under uncertainty (e.g. see [32]), only [26], [34], [35] have considered the use of arguments for computing trust in a local trust rating setting. Reference [34] proposes an argumentation-based approach for trust evaluation that is bipolar (separating arguments for trust and for distrust) and qualitative (as arguments can support various degrees of trust/distrust). Reference [35] derives argumentation logic where arguments support measures of trust, e.g. qualitative measures such as "very reliable" or "somewhat unreliable". There are several non-argumentation based methods to model the trust/distrust of the evaluator in the target [36]. Reference [10] classify approaches to trust as either "cognitive", based on underlying beliefs, or "game-theoretical", where trust values correspond to subjective probabilities and can be modeled by uncertainty values. However, [37] argue against a purely game-theoretic approach to trust and in favour of a cognitive approach based upon a mental model of the evaluator, including goals and beliefs. Moreover, some works (e.g. [38]) advocate the need for and benefits of hybrid trust models [36], [39], combining both the cognitive and game-theoretical approach.

1) Extending the Trust Model using Argumentation

It is important to add argumentation to the above trust model (as described in section 3) based on only acceptable and unacceptable recommendations as interactions. Arguments are important for giving the explanation behind any kind of interaction between the agents. This improves the quality and utility of recommendation iteratively, that is over several cycles [16]. During argumentation, the agents may agree or disagree over certain issues before finally the user either accepts or rejects a recommendation. We believe that whatever may be the eventual result of a recommendation process, always various arguments (in support or against) are responsible for it. This is because these arguments form the basis for the generated recommendations. Therefore, it is vital to determine the agents' responses on such arguments besides recommendations. This would help in determining a more accurate trust value for agents in the system.

Therefore, to evaluate the trustworthiness of an agent y, an agent x uses the history of its interactions (both arguments and recommendations) with an agent y. Eq. 6 shows how to calculate this fuzzy measure of trustworthiness [26].

$$TRUST_{xy} = \frac{ag(Arg_{xy}) - disag(Arg_{xy}) + R_{xy} * w_{satisfied} - R_{xy} * w_{unsatisfied}}{T\_N\_Arg_{xy} + T\_N\_R_{xy}} \quad (6)$$

where $TRUST_{xy}$ denotes the trustworthiness of y according to x's point of view. Due to lack of space we use notational representation for eq. (6). Therefore,

$ag(Arg_{xy})$ represents $\sum_{Arg \in I} \text{FM\_agree}(Arg_{xy})$, that is the summation of the fuzzy measure of the degree of x's agreement over y's arguments that are acceptable to x; obtained from eq. (11).

$disag(Arg_{xy})$ represents $\sum_{Arg \in I} \text{FM\_disagree}(Arg_{xy})$, that is the summation of the fuzzy measure of the degree of x's disagreement over y's arguments that are unacceptable to x; obtained from eq. (12).

$R_{xy} * w_{satisfied}$ represents $\sum_{R \in I} FM(R_{xy}) * w_{satisfied}$, which is defined as described in eq. (1).

$R_{xy} * w_{unsatisfied}$ represents $\sum_{R \in I} FM(R_{xy}) * w_{unsatisfied}$, which is defined as described in eq. (1).

$T\_N\_Arg_{xy}$ is the total number of arguments made by y towards x; a count maintained by the agent's persistent belief base in the system.

The possible combinations of satisfied and unsatisfied fuzzy subsets define four values for any two agents. For argumentation between agents in a recommender system we need to define the following two combinations only, i.e. satisfied-satisfied written as SS(x, y), and unsatisfied-satisfied written as US(x, y), assuming that agent x represents a user whereas agent y represents a recommender.

$$SS(x,y) = \frac{|satisfied\ (x) \cap satisfied\ (y)|}{|satisfied\ (x) \cup satisfied\ (y)|} \quad (7)$$

$$US(x,y) = \frac{|unsatisfied(x) \cap satisfied\ (y)|}{|unsatisfied(x) \cup satisfied\ (y)|} \quad (8)$$

Fuzzy sets literature describes many alternatives for union and intersection of crisp sets. The popular one is minimum for intersection and maximum for union. The min–max alternative and the definition of fuzzy set's cardinality as given by Zadeh in [40] yield the following:

$$|\,\text{satisfied}(x) \cap \text{satisfied}(y)\,| = \sum_{i_k \in H_x \cap H_y} \min(\text{sat}_x(i_k), \text{sat}_y(i_k)) \quad (9)$$

$$|\,\text{satisfied}(x) \cup \text{satisfied}(y)\,| = \sum_{i_k \in H_x \cap H_y} \max(\text{sat}_x(i_k), \text{sat}_y(i_k)) \quad (10)$$

Therefore, for a rating fuzzy trust system, the agreement and disagreement values over an argument between any two agents x and y are given by:

$$\text{FM\_agree}(Arg_{xy}) = SS(x,y) \quad (11)$$

$$\text{FM\_disagree}(Arg_{xy}) = US(x,y) \quad (12)$$

B. Trust for Argumentation

During an argumentation process there is an exchange of arguments that are built using different pieces of knowledge. Whether an argument is finally accepted or not depends, amongst other things, on the structure of the argument and the "truth" behind the knowledge used to build the argument. More than often, the "truth" of the knowledge that is used to build the arguments may depend on the source of that knowledge. The stream of information received from an informant together with its trust or reputation are the only elements an agent (or human user) can use to decide whether an argument, built from that information, can be accepted or not. The use of trust models for this purpose is straightforward. Currently, one of the main uses of trust models in multi-agent systems is the evaluation of a piece of information regarding the source (who is the origin) of that information. Extending this kind of evaluation to each one of the elements of a (structurally correct) argument, we could evaluate the truthfulness of that argument. In an agent-based system there are arguments related to an agent's mental attitudes (beliefs, desires and plans). To deal with the different nature of the arguments involved, we have developed three distinct argumentation frameworks: one for reasoning about beliefs, another for arguing about what desires should be pursued, and a third for arguing about the best plan to intend in order to achieve these desires. In our present work, we focus only on the strengthening of argumentation for planning by using trust and in turn improving our trust model as well.

1) Argumentation Based Planning with Trust

Planning is a substantial and well-developed area in AI [41]. Our aim is not to propose a novel planning framework here. Instead, we intend to use the notion of trust between agents to improve strength of an instrumental argument required to build a plan. An instrumental argument (a necessary sub-goal for realization of a super-goal) may achieve one or several desires of different worth (certainty and preference) required for the complete execution of a recommendation plan. So the strength of that argument is the "benefit" or "utility" which is the cumulative worth of the desires (sub-goals) essential in the realization of the plan. $\mathcal{D}$ denotes the set of desires, a base B_p contains planning rules and $\mathcal{RES}$ denotes a set of resources available in the system. We modify the way *strength or weight* of instrumental arguments were calculated initially [27], so as to take account of trust (on the source of plan) besides user's preference in determining the best plan. After defining the basic building block for specifying plans that is the notion of planning rule (refer definition 5 in section 2), we now define the concept of partial and complete plans.

Definition 8 (Partial plan). *A* partial plan *is a pair* $[H, \varphi]$ *where*

- $\varphi \in R$ *and* $H = \Phi$ *, or*
- $\varphi \in D$ *and* $H = \{\varphi_1, \ldots, \varphi_n, r_1 \ldots, r_m\}$ *such that* $\exists\ \varphi_1 \wedge \ldots \wedge \varphi_n \wedge r_1 \ldots \wedge r_m \rightarrowtail \varphi \in B_p$.

A partial plan $[H, \varphi]$ *is* elementary *iff* $H = \Phi$.

Definition 9 (Instrumental argument, or complete plan for recommendation). *An* instrumental argument *is a pair* $<T, d>$ *such that* $d \in D$, *and T is a finite tree such that*:

- *the root of the tree is a partial plan* $[H, d]$;
- *a node* $[\{\varphi_1, \ldots, \varphi_n, r_1 \ldots, r_m\}, h']$ *has exactly* $n + m$ *children* $[H'_1, \varphi_1], \ldots [H'_n, \varphi_n], [\Phi, r_1], \ldots [\Phi, r_m]$ *where each* $[H'_i, \varphi_i]$, $[\Phi, r_k]$ *is a partial plan;*
- *the leaves of the tree are elementary partial plans.*

To collect all elements of an instrumental argument: Nodes(T) *is a function which returns the set of all partial plans of tree T, Des*(T) *is a function which returns the set of desires that plan T achieves, and Resources*(T) *is a function which returns the set of all resources needed to execute T.*

Let $\mathcal{A}_p$ denotes the set of all instrumental arguments (for various recommendation plans) that can be built from agent's bases. An instrumental argument (a necessary sub-goal for realization of a super-goal) may achieve one or several desires of different worth required for the complete execution of a recommendation plan generated by a recommender agent in the system. Therefore, to determine the best plan to intend (instrumental argument with maximum strength), the strength of that argument is determined by the trust value in the agent (origin of the argument) along with its utility (which is the cumulative worth of the desires (sub-goals) essential in the realization of the plan). Formally:

Definition 10 (Strength of instrumental arguments). *Let A* $= <T, d>$ *be an instrumental argument,* **'w'** *represent its strength. The* utility *of A is given as*

$$\text{Utility}(A) = \sum_{d_i \in Des(T)} Worth(d_i) \tag{13}$$

Therefore, strength of the argument, i.e.,

$$\boldsymbol{w = TRUST_{xy} * Utility\ (A)} \tag{14}$$

Equation (14) is obtained using equation (6) and equation (13). For definition of *worth of desires,* refer definition 6 in section 2. Note that, here agent 'x' is an 'evaluator' and agent 'y' is the 'target'.

In [42], it has been shown that there are four families of conflicts between partial plans. In fact, two partial plans $[H_1, \varphi_1]$ and $[H_2, \varphi_2]$ may be conflicting for one of the following reasons:

- *desire-desire* conflict, *i.e.* $\{\varphi_1\} \cup \{\varphi_2\} \vdash \perp$.
- *plan-plan* conflict, *i.e.* $H_1 \cup H_2 \vdash \perp$.
- *consequence-consequence* conflict, *i.e.* the consequences of achieving the two desires h_1 and h_2 are conflicting.
- *plan-consequence* conflict, *i.e.* the plan H_1 conflicts with the consequences of achieving h_2.

The above conflicts are captured when defining the notion of *conflict-free* sets of instrumental arguments. Using *definition 10*, these conflicts can now be resolved in favor of an argument from a trustworthy source.

Argument selection is the essence of the argumentation-based recommendation strategy. This mechanism consists of selecting one argument to be uttered from among the set of candidate arguments that might be uttered. The agent selects an argument on the basis of the argument strength determined by preference, its trust in the other agent or both. Thus, if there are two different candidate arguments, the agent will select one by applying one of those policies to determine which one should be uttered. In this context, the action (to be executed) selection mechanism of the planning algorithm must take into account the argument selection policy of the agent in order to generate argumentation plans (instrumental arguments). In our work, we represent this policy into the agent's mental state as its trust in other agents and its preferences for actions and goals.

V. Integrating Trust with Argumentation Plan Generation

Once the mechanism to construct argumentation plans [43] and trust is defined, we can integrate this construction into the general planning of the agent. The agent plans the course of action that it must follow to achieve its goals [44]. This plan may be composed of several actions or sub-goals. Some of these sub-goals or goals are under the direct control of the agent. Other goals are not under the control of the agents. The latter must be performed cooperatively. The agent usually builds a plan and starts to execute it by stopping at every non-controlled goal or action (requiring a recommendation, consultation or some information) in order to get it executed by some of the agents that can do this action. However, whenever the recommendation fails, in the least expensive case, the agent must find another alternative to reach the same goal. In the most expensive case, the agent must re-plan its course of action. This happens because when planning the course of action, the agent did not take into account trust factor and planning for alternatives as a critical goal (action) within the general plan. In this situation, both the failed recommendation and the actions executed before it, which cannot be reused, consume resources (in terms of execution time and the communication messages passed).

At this point, the argumentation plans (instrumental arguments) with trust can play a fundamental role. If the agent has an argumentation plan, which indicates the necessary arguments, to agree on the execution of a recommendation action, it will have a good hint to trust the viability of the recommendation. Therefore, if we integrate the construction of these proposed plans into the construction of the general plan, the agent will have a clear vision of the dialogues that it should carry out, with whom it should seek recommendation (for a trustworthy source) and which arguments it should utter to reach its goal. Thus, the impossibility of reaching its goal can be detected at an early stage (planning time), and the agent may modify its general plan, without the need to completely execute it and waste resources.

The integration can be easily accomplished. On the one hand, we have the definitions of the initial state (i.e. agent seeks recommendation), final state (i.e. agent accepts a recommendation) and actions (goals and sub-goals) of the general planning problem. On the other hand, we need the same definitions for each argumentation plan required for each alternate recommendation action that is added to the general plan. The initial state of the general plan includes the information that the agent has about the world in which it is performing. Additionally, the general actions, which are settled before the argumentation plan, modify the world (effects) in which the argumentation plan will be executed. Therefore, these actions also contribute to form the implicit initial state of the argumentation plan.

Figure 3 shows a view of the plan. This figure illustrates an initial state which is implicit, because it is not defined explicitly by the agent, but it is created during the general plan construction. Something similar occurs with the final state of the argumentation plan. No intermediate final state is explicitly defined, but the goals or actions which are not under the control of the agent have a special precondition that forces the planning algorithm to build such a plan. The precondition about trust is added, and it implicitly represents the final state of the argumentation plan i.e. the goal of obtaining trustworthy recommendation, which the agent wants to reach. Besides adding the new precondition to the actions, it is necessary to have an alternative to the argumentation plan when actions of the general plan cannot be supported by one of these plans and when substitute actions (goals) or other trusted agents do not exist.

In other words, there are situations in which an argumentation plan that agrees on the execution of a necessary goal cannot be built, either because there is not sufficient information or because there is no evidence indicating that the recommender agent, from whom the user agent should request such execution, is trustworthy enough. In these cases, we cannot claim that there is not a viable general plan, only that there is not an argumentation plan. Therefore, in order not to deprive the planning algorithm of the possibility to build a valid general plan in these situations, we maintain one unconditional action for each existing argumentation plan. These unconditional actions (of simply accepting or rejecting a recommendation) are included in the initial definition of the problem, without the precondition of trust. We also add preferences that prioritize the usage of argumentation plan over general ones to give a fair chance to the former one. Thus, the planning algorithm first searches for the actions that can be supported by an argumentation plan using trust, but if this plan cannot be built, the algorithm includes the unconditional actions that must be requested and executed traditionally for a RS (Recommender System).

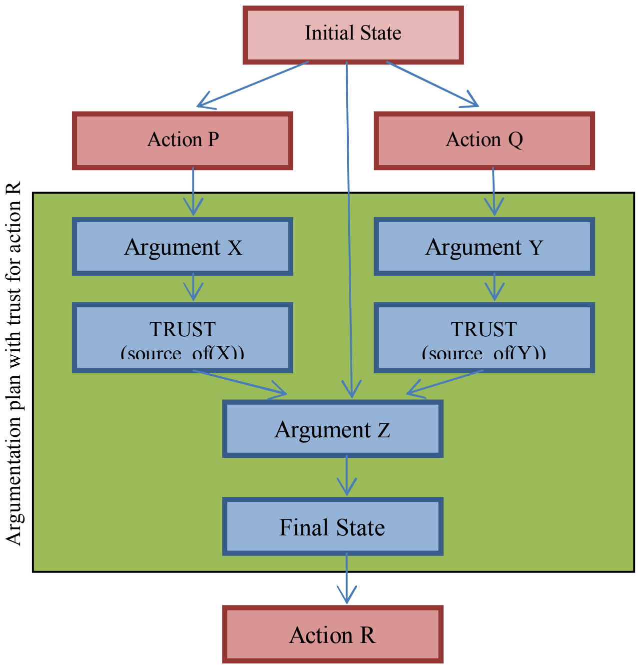

Figure 3. A view of the plan

To formalize the plan construction, Figure 4 shows the algorithm that allows the agents to build a general plan. Initially, the agent, which must achieve a certain goal (step 2), defines initial states, final states, the actions and the preferences (trust and utility) as defined above (step 4-7). Then, the agent builds a general plan (step 8). If the planning algorithm can build a plan, the agent tries to execute it (step 13); otherwise the agent waits for a change in the context that allows it to build a new plan (step 15). During the execution of the plan three situations may occur: (a) the plan is executed successfully; (b) an argumentation plan fails, then the agent can continue recommendation process traditionally and follow the general plan; or (c) the general plan fails, then the agent must re-plan the course of action, due to the fact that its goals remain unachieved. After the plan execution, the agent checks if its entire goals have been fulfilled (step 2). If so, the agent finishes the execution and stays on standby until new goals appear. On the other hand, if there are unachieved goals, the agent needs to obtain a new plan. To do this, the agent can request a new plan from the planning algorithm (re-planning, step 10) if the context information has not changed (this method has no arguments because the re-planning is performed using the same information used to plan). Otherwise, a new plan must be built, since the initial definition (specifically, the initial state) is no longer valid (step 4-8). Notice that the re-planning time can be improved if the re-planning process is carried out concurrently with the execution of the original plan. Thus, if the original plan fails, the re-planning process could be able to build a re-plan more quickly.

```
1. condition_change : = TRUE;
2. WHILE (require recommendation OR goals
     unachieved) DO
3.       IF (condition_change) THEN
4.             A : = {actions U arguments};
5.              I : = {initial state};
6.              F : = {final state};
7.              TUP : = {trust U utility U preferences};
8.              plan : = plan_generation (A, I, F, TUP);
9.               ELSE
10.                  plan : = plan.replanning ();
11.              END IF
12.              IF (plan < > Φ) THEN
13.                   plan.execute ();
14.              ELSE
15.                   wait_for_condition_change ();
16.             END IF
17.      condition_change : = check_for_change (A, I, F,
     TUP);
```

Figure 4. Algorithm for plan construction

VI. Experiments and Results

The recommendation scenario presented here, can provide alternatives to obtain the resources needed to achieve the agents' goals within a single planning iteration. It is also feasible to say that, several alternatives exist in a real scenario. These alternatives facilitate the achievement of the goals, but also make the decision-making process complex. Moreover, not all alternatives are viable or even trustworthy. Therefore, any mistake in the decision-making process could lead to reduced utility, lack of time and resources and make the goals unattainable. To show how our proposal works in such situations, we have multiple user and recommender agents in the recommendation scenario set up for an agent-based recommender system using trust and argumentation. Experiments are based on a case study for book recommendations. The system uses the original book dataset which has been elicited from http://csl.du.ac.in/. This is the official website of Delhi University for its Central Science Library (CSL), which is a huge repository of thousands of books for several academic disciplines. Presently, we concentrated only on the computer science books for building a book dataset for the prototype. CSL provides a book issue facility for 6507 titles under the discipline 'Computer Science'. For the experimental study, 30 different relevant user profiles from various categories were created, who actually rated over 1000 books on the four selected book attributes (author, publisher, publication year (oldness) and cost). This system was developed using Jason for building agents enabled with inference and interaction capabilities. These agents have BDI architecture which is suitable for such a set up. Each recommender agent has one goal: to get its recommendation accepted. In this context, user agent must select the recommender agent that it can trust in order to obtain the recommendation that it needs to fulfill its goal (of obtaining a book according to its preferences). Hence, the recommender agents will try to persuade user agent by their arguments. An agent may accept, reject, assert or counter-attack an argument from another agent. The stronger argument will emerge as a winner.

The evaluation of this new scenario was carried out in an incremental way. That is, we added alternative resources one by one. We calculated five metrics in each situation, for measuring the performance of agents in the system. These metrics are the following:

- ***T***: total time taken by user to fulfill the goal.
- **#*I***: total times that user initiated a recommendation.
- **#*P***: number of times that the planning algorithm was executed since an initial stage. This happens when the agent execution starts or when the world changes after a failed plan execution and a new solution cannot be requested from the planning algorithm.
- **#*Rp***: total times that the agent requested a new plan solution from the planning algorithm (replanning). After a failed plan execution, the agent must solicit a new plan solution. If the world (user's requirement for recommendation) does not change, the planning problem continues being valid and the algorithm will give a new solution.
- **#*M***: total messages exchanged by all agents during the execution.

The main goal of this evaluation was to compare our proposal with the traditional recommendation process and argumentation planning with and without trust. On the one hand, we evaluated the performance of agent with the ability of building argumentation plans. On the other hand, we also evaluated the performance of the same agent without such ability. We call the agent using traditional simple way of recommendation process as *Agent_sim,* the one using argumentation planning as *Agent_arg* and the agent using argumentation planning with trust as *Agent_argtrust.* The comparative results are shown in Table 1, where #n_i is the number of alternatives n_i available (alternative resources' recommendation for achieving a goal) and included in the execution. As shown in Table 1, *Agent_sim's* performance is better in the first two executions (when lesser alternatives were available), where it fulfilled its goal in a shorter time, but required two executions of the planning algorithm. However, in *Agent_arg*'s plan there is a relation between the argumentation plans that support its actions. Due to lesser alternatives available and lack of user's agreement over the available recommendations, the plan for recommendations fails because *Agent_arg* has no more information to build arguments for alternatives. In execution #n_2, *Agent_sim*'s performance continues being better than *Agent_arg*'s and *Agent_argtrust*'s performances, but time *T* of the second execution is more than twice the time of the first one. This happens because the planning algorithm selects in a non-deterministic way, an agent for providing recommendation. Hence, in this case it needs to request a new solution from the planning algorithm when one fails. The new solution is an alternative plan.

Table 1. Comparison of Agent_Argtrust, Agent_Arg and Agent_Sim Performances.

#n_i	*Agent_argtrust*					*Agent_arg*					*Agent_sim*				
	T	*#I*	*#P*	*#Rp*	*#M*	*T*	*#I*	*#P*	*#Rp*	*#M*	*T*	*#I*	*#P*	*#Rp*	*#M*
1	01'04"	3	1	0	53	01'03"	3	1	0	53	00'24"	7	2	0	95
2	01'18"	3	1	0	65	01'10"	3	1	0	67	00'58"	10	2	1	113
3	01'25"	3	1	0	60	01'27"	3	1	0	65	01'29"	18	3	2	169
4	01'38"	3	1	0	57	01'38"	3	1	0	79	02'19"	20	3	3	170
5	02'10"	3	1	0	71	02'19"	3	2	1	86	02'43"	19	2	4	168
6	02'36"	3	1	0	80	02'38"	3	1	0	96	03'40"	22	2	5	188
7	02'40"	3	1	0	77	02'59"	3	2	1	99	05'29"	25	2	6	197
8	02'53"	3	1	0	65	03'28"	3	2	1	105	06'39"	32	4	7	220
9	03'00"	3	1	0	69	04'04"	3	1	0	114	08'57"	31	2	8	227
10	03'13"	3	1	0	74	04'58"	3	1	0	122	11'20"	38	5	9	239

From the execution #n_3 onwards, when the number of available alternatives starts increasing, *Agent_arg* starts obtaining better results than *Agent_sim* whereas, *Agent_argtrust* is consistently showing slightly better performance as compared to *Agent_arg*. *Agent_argtrust* is better because of its trust determination capability amongst the available alternatives.

Figure 5(A) compares the time taken by *Agent_argtrust, Agent_arg* and *Agent_sim* to achieve their goals. Axis *X* represents the number of alternatives n_i available (alternative recommendation for achieving a goal) and included in the execution. Axis *Y* represents the time taken by the agent. As can be seen, the time taken by *Agent_argtrust* is lesser than the time taken by *Agent_arg*. It can be seen that the time taken by *Agent_argtrust* and *Agent_arg* both, increases almost linearly, because the agents decide from whom to seek recommendation and evaluates options during planning time, thus avoiding possible failures in the plan execution. In contrast, for *Agent_sim*, the time increases exponentially, because it does not take into account the evaluation of alternatives and trust within the planning process. Therefore, *Agent_sim* cannot determine in planning time with which agent it is better to move ahead in the process. Thus, the time needed to achieve the goals increases exponentially because the agent has to execute the planning algorithm several times and requires several plan solutions until it finds the successful one. Moreover, the number of messages uttered by the agents when *Agent_sim* is participating is meaningfully higher than when *Agent_argtrust* is participating. The reason for this difference lies in the different executions and failures of the plans, which lead to an excessive number of messages exchange. Figure 5(B) shows the total messages uttered by the agents while *Agent_sim, Agent_arg* and *Agent_argtrust* are participating in the recommendation process. As these figures indicate, there is a reduction not only in the execution time of the argumentation plan generation (with trust), but also consequently, in the overload of the communication channel shared by the agents.

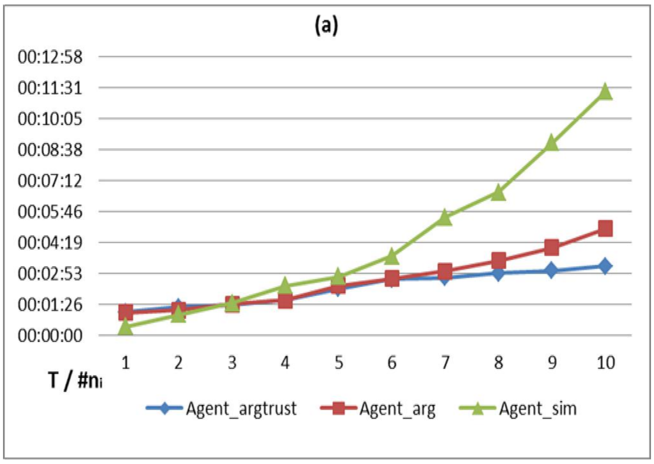

Figure 5(A). Comparative charts between *Agent_sim, Agent_arg* and *Agent_argtrust* performances.

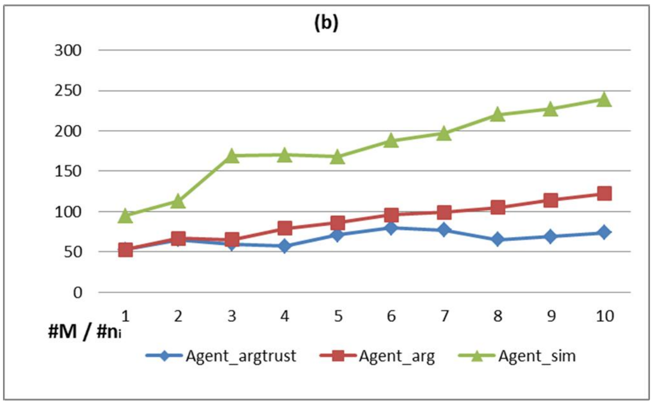

Figure 5(B). Comparative charts between *Agent_sim, Agent_arg* and *Agent_argtrust* performances.

VII. Discussions

As shown in the previous section, one of the main contributions of our approach is that it allows us to integrate the recommendation in general, and the argumentation with trust in particular, within the planning stage. This integration allows the agent to take trustworthy decisions in advance, in planning time, before the execution of the general plan is carried out. Recommendation systems try to assist users during the different recommendation stages and convince them over a product [5], [45]. Agent technology has also been integrated within these systems to model many decision-making tasks requiring recommendation, especially since these agents are an excellent tool to assist users or to allow them to act on behalf of users [4], [7], [29], [46]. RSs have also been built based on social factors related to user's personality and trust to improve the satisfaction of users involved in the process [3], [4]. Our proposed approach can be applied in both the directions. That is, on the one hand, the agent can act on behalf of a user, taking into account his/her preferences, goals, plans and trust issues to support decision making in situations where the source of the information on which decisions are based is of varying trustworthiness [12]. In this case, the agent will provide and improve recommendations autonomously. On the other hand, a personal agent can assist a user during the recommendation process by using the information extracted (due to argumentation) from the integral plans. That is, a user operating a recommendation system can receive assistance to make decisions from a personal agent. In this sense, we distinguish among several decisions in which the information taken from the integral plans can be applied:

- To decide with which agent we must seek recommendation in case where there are several alternatives to choose from.
- To evaluate in advance the need to reach secondary alternatives available.
- This leads to increase in trust in a RS as higher number of arguments (finally uttered after planning) are getting accepted and user is more satisfied with the recommendation process due to change in the planning process.

Finally, as the results have shown, making early decisions at this point allows the user and the recommender to save effort, time and resources.

VIII. Conclusion and Future Work

Argumentation technologies are promising tools for the settings where autonomous agents can support humans in decision making. Agents can help their users in identifying the most profitable choice (recommendation) of all to take a decision accordingly. In this paper, we have considered the integration of trust mechanism in argumentation based recommender system. Especially, in an argumentation-based recommendation system, the arguments uttered to persuade each other over a product are not the result of an isolated analysis, but of an integral view of the problem that we want to agree about. Agents plan the actions that they should execute to achieve their goals. Furthermore, the planning algorithm utilizes argumentation and the agent's trust preferences in order to select the best actions. In this work, we have used argumentation for handling trust and vice versa. Trust in an individual is important when the trusting party needs that individual to perform an action for them. The feature of using influence of trust on argumentation provided a useful versatility to the problem of planning for autonomous agents. This integration allowed the user to take well-reasoned decisions based on trustworthy recommendations. This improves performance of the agents in terms of the communication overhead caused (number of messages passed) and time taken for decision making. The same was confirmed by the results obtained from experiments conducted for a Book Recommender System (RS).

As part of our future work, we are working on the integration of trust with the argumentation frameworks for arguing about beliefs and desires. We are also studying the effects of such integration on an agent's practical reasoning in a recommendation scenario.

References

[1] F. Abel et al., “Augmenting User Models with Real World Experiences to Enhance Personalization and Adaptation”, Advances in User Modeling, LNCS, Vol.7138/2012, 31-34, 2012.

[2] F. Grasso, J. Ham and J. Masthoff, “User Models for Motivational Systems, The Affective and the Rational Routes to Persuasion”, LNCS, Advances in User Modeling,Vol.7138/12,pp.335-341,2012.

[3] L. Q. Sanchez, J. A. R. Garcia, B. D. Agudo, G. J. Diaz. “Social factors in group recommender systems”, *ACM Transactions on Intelligent Systems and Technology (TIST),* Volume 4 Issue 1, January 2013, Article No. 8, ACM, NY, USA, 2013.

[4] J. A. R. Garcia, L. Q. Sanchez, B. D. Agudo. “Including Social Factors in an Argumentative Model for Group Decision Support Systems Supported”, *Decision Support Systems,* In Press, Accepted Manuscript, Available online 27 May 2013.

[5] K. H. Yoo, U. Gretzel, M. Zanker. *Persuasive Recommender Systems: Conceptual Background and Implications,* Springer, 2013.

[6] A. Felfernig, B. Gula, G. Leitner, M. Maier, R. Melcher and E. Teppan, “Persuasion in Knowledge-Based Recommendation”, H. Oinas-Kukkonen et al. (Eds.): PERSUASIVE 2008, LNCS 5033, pp. 71–82, Springer-Verlag, 2008.

[7] C. Chesnevar, A. G. Maguitman and M. P. Gonzalez, “Empowering Recommendation Technologies Through Argumentation”, In: I. Rahwan, G. Simari (Eds.), Argumentation in AI, Springer-Verlag, 2009.

[8] A. Jøsang, R. Ismail and C. Boyd, “A survey of trust and reputation systems for online service provision”, Decision Support Systems, Vol. 43, pp. 618– 644, Elsevier, 2007.

[9] D. Artz and Y. Gil, “A survey of trust in computer science and the semantic web”, Journal of Web Semantics, 5(2):58–71, June 2007.

[10] J. Sabater and C. Sierra, “Review on computational trust and reputation models”, Artif. Intell. Rev, 24(1):33-60, 2005.

[11] P. Bedi, H. Kaur and S. Marwaha, “Trust based recommender system for the semantic web”, In proceedings IJCAI'07 - International Joint Conference on Artificial Intelligence, January 9-12, 2007, Hyderabad, 2677- 2682. USA: AAAI Press, 2007.

[12] S. Parsons, E. Sklar, M. Singh, K. Levitt, J. Rowe. “An Argumentation-Based Approach to Handling Trust in Distributed Decision Making”, *Trust and Autonomous Systems: Papers from the 2013 AAAI Spring Symposium*, pp. 1951-1957, 2013.

[13] X. L. Dong, L. Berti-Equille and D. Srivastava, “Integrating conflicting data: The role of source dependence”, In Proceedings of the 35th International Conference on Very Large Databases, Lyon, France, August 2009.

[14] Y. Tang, K. Cai, P. McBurney, E. Sklar. “Using argumentation to reason about trust and belief”, *Journal of Logic and Computation,* Oxford Journals, Mathematics & Physical Sciences, Volume 22, Issue 5, pp. 979-1018, 2012.

[15] S. Parsons, E. Sklar, P. McBurney. “Using Argumentation to Reason with and about Trust”, *Argumentation in Multi-Agent Systems, Lecture Notes in Computer Science,* Volume 7543, pp. 194-212, Springer, 2012.

[16] P. Bedi and P. Vashisth, “Improving Recommendation by Exchanging Meta - Information”, In proceedings CICN 2011 Oct 07- 09, MIR Labs, Gwalior, India, IEEE, 2011.

[17] P. Vashisth, D. Chandoliya, B. Kr. Yadav and P. Bedi, “Trust enabled Argumentation Based Recommender System”, In proceedings ISDA 2012 - 12th International Conference on Intelligent System Design and Applications, November 27-29, 2012, Kochi, India: 137-142, USA: IEEE Xplore., 2012.

[18] P. Maes, R. Guttman and A. Moukas, “Agents that buy and sell”, Commun. ACM, 42 (3), pp. 81–ff, 1999.

[19] M. Maatallah and H. Seridi, “A fuzzy hybrid recommender system,” Machine and Web Intelligence (ICMWI), 2010 International Conference, IEEE, pp. 258-263, 3-5 Oct. 2010

[20] Y. Cao and Y. Li, “An intelligent fuzzy-based recommendation system for consumer electronic products,” Expert Systems with Applications: An International Journal, Vol. 33, Issue 1, pp. 230 - 240, 2007.

[21] H. Stormer, N. Werro and D. Risch, “Recommending Products with a Fuzzy Classification,” CollECTeR Europe 2006.

[22] R. R. Yager, “Fuzzy logic methods in recommender systems,” Fuzzy Sets and Systems, Vol. 36, Issue 2, pp. 133-149, 2003.

[23] H. C. Tseng, “Internet applications with fuzzy logic and neural networks: A survey,” Computer Science Dept., San Jose State University, CA, USA, Journal of Engineering and Computing and Architecture, Vol. 1, ISSN 1943-7197, Issue 2, 2007.

[24] S. Xiaowei, “An intelligent recommender system based on fuzzy logic,” Informatics In Control, Automation and Robotics I, Springer Netherlands, Part 2, pp. 105-109, 2006.

[25] K.K. Bharadwaj and M.Y.H. Al-Shamri, “Fuzzy computational models for trust and reputation systems”, Electron. Comm. Res. Appl., doi:10.1016/j.elerap.2008.08.001, 2008.

[26] P. Bedi and P. Vashisth, “A Fuzzy Trust Model for Argumentation-Based Recommender Systems”, International Conf. on SocPros 2011, IIT Roorkee, INDIA, Springer, 2011.

[27] P. Bedi and P. Vashisth, “Interest Based Recommendations with Argumentation”, Journal of Artificial Intelligence. ANSI,4:119-142,DOI:10.3923/jai.2011.119.142, 2011.

[28] A. Felfernig, M. Mairitsch, M. Mandl, M. Schubert, and E. Teppan, “Utility-Based Repair of Inconsistent Requirements”, B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 162–171, Springer-Verlag, 2009.

[29] C. Chesñevar, A. Maguitman, E. Estevez, R. Brena. “Integrating argumentation technologies and context-based search for intelligent processing of citizens' opinion in social media”, *In Proceedings of ICEGOV '12, ACM,* NY, USA, pp. 166-170, 2012.

[30] I. Rahwan and L. Amgoud, "An argumentation based approach for practical reasoning", In: N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 74–90, Springer-Verlag, 2007.
[31] G. Boella et al., "Arguing about Trust in Multiagent Systems", 11th Symposium on Artificial Intelligence of the Italian Association for Artificial Intelligence, 2010.
[32] S. Parsons et al., "Some thoughts on using argumentation to handle trust", (Invited Talk) CLIMA 2011: 1-12, 2011.
[33] Y. Tang et al., "A system of argumentation for reasoning about trust", In Proceedings of the 8th European Workshop on Multi-Agent Systems, Paris, France, 2010.
[34] H. Prade, "A qualitative bipolar argumentative view of trust", In SUM, 2007.
[35] S. Parsons et al., "Argumentation-based reasoning in agents with varying degrees of trust", Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS, pp. 879-886, 2011.
[36] P. Victor, C. Cornelis, M. De Cock and A.M. Teredesai, "Trust and distrust-based recommendations for controversial reviews", IEEE Intelligent Systems, 2012.
[37] C. Castelfranchi and R. Falcone, "Trust is much more than subjective probability: Mental components and sources of trust", In HICSS, 2000.
[38] E. Staab and T. Engel, "Combining cognitive with computational trust reasoning", In AAMAS-TRUST, volume 5396 of LNCS. Springer, 2008.
[39] P. A. Matt, M. Morge and F. Toni, "Combining statistics and arguments to compute trust", In W. van der Hoek and G. A. Kaminka, editors, Proceedings of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), 2010.
[40] L.A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages", Computer and Mathematics with Applications 9 (1), pp.149–184, 1983.
[41] J. A. Baier and S. A. McIlraith, "Planning with preferences", AI Magazine, 29 (4), pp. 25-36, 2008.
[42] L. Amgoud, "A formal framework for handling conflicting desires", In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 552–563. Springer, Heidelberg, 2003.
[43] A. Monteserin and A. Amandi, "Argumentation-based negotiation planning for autonomous agents", Decision Support Systems, 2011.
[44] L. Berdún and A. Amandi, "Planning for intelligent agents", In Proc. of the 7th Argentine Symposium on Artificial Intelligence, Rosario, Argentina, pp. 12-23, 2005.
[45] P. Bedi et al., "A Website Recommender System based on Analysis of User's Access Log", Journal of Intelligent Systems, Special Issue on Recent Advances in Artificial Intelligence. 18(4): 333-352, 2009.
[46] J. Lang, M. Spear and S. F. Wu, "Social manipulation of online recommender systems", In Proceedings of the 2nd International Conference on Social Informatics, Laxenburg, Austria, 2010

Author Biographies

Dr. Punam Bedi was born in Delhi, India in 1962. She has done her M.Tech(CS) from IIT Delhi and was awarded a Doctorate in computer science in 1999 by University of Delhi, India. She is an Associate Professor in Computer Science, University of Delhi. She has 27 years of teaching and research experience, and has published about 100 research papers in International Journals. Her research interests include Multi-agent Systems, Personalization, Intelligent Information Systems, Intelligent User Interfaces, and Human Computer Interaction (HCI).

Pooja Vashisth was born in Delhi, India in 1979. She completed her M.Tech(IT) in 2006 from Guru Gobind Singh Indraprastha University, Delhi, India. She is presently pursuing her PhD in Computer Science from University of Delhi. She is an Assistant Professor in Computer Science, SPM College, University of Delhi with a teaching experience of 12 years. Her research interests include Multi Agent Systems, argumentation, user modeling, and recommender systems.