

Pro-Active Multi-Dimensional Recommender System using Multi-Agents

<http://dx.doi.org/10.3991/ijim.v6i3.2012>

H. Al Tair¹, M. J. Zemerly¹, M. A. Al Qutayri¹ and M. Leida²

¹ Khalifa University, Sharjah, UAE

² Etisalat-BT Innovation Centre (EBTIC), Abu Dhabi, UAE

Abstract—recommender systems currently used in many applications, including tourism, tend to simply be reactive to user request. The recommender system proposed in this paper uses multi-agents and multi-dimensional contextual information to achieve proactive behavior. User profile and behavior get implicitly incorporated and subsequently updated in the system. The recommender system has been developed and applied to the tourism domain. It was tested and evaluated by relatively large set of real users. The evaluation conducted shows that most of the users are satisfied with the functionality of the system and its ability to produce the recommendation adaptively and proactively taking into considerations different factors.

Index Terms—Multi-agent, multi-dimensional rating, pro-activity, recommender system.

I. INTRODUCTION

Recommender systems are normally used to filter huge amounts of information in order to provide users with recommendations pertaining to products/services. These recommendations are usually filtered based either on the information content or ratings by other users. Systems using the former approach are called Content Based Filtering (CBF) while those that utilize the latter are called Collaborative Filtering (CF) [put refs]. These two approaches are viewed as the traditional recommender systems since they were introduced first and have been applied in many recommender systems by product/service providers such as Amazon¹ and Netflix². Each of the said traditional approaches has its own limitations as will be discussed in Section 3 [1]. The traditional approaches are not suitable for use in context aware situations. In such situations recommendations depend on several different conditions that can vary from one a case to another. For example, in the tourism field, the target users of the recommender system would have different preferences and it is not necessary that they all would be in the same touristic areas that may be recommended [2]. Hybrid recommender systems overcome such shortcomings of the traditional approaches [1]. Knowledge-based is an example of the hybrid recommender system types [3]. This approach works based on previously defined knowledge or rules. The benefit of having rules is that they can be used for critical situations and hence have the tasks done more effectively and in harmony with the user context.

To enhance the performance of two dimensional recommender systems a number of researchers proposed

integrating them with multi-agent systems [4] and [5]. A major feature of agent technology is that it is based on a collaborative model, especially among agents, with other components of the system. This aspect can be used to facilitate the process of arriving at appropriate recommendations. Another property of agents is that they can be designed in order to be pro-active. This means they have the capability to initiate and execute tasks autonomously.

Most of the above recommender systems are considered as two dimensional (the user and the product/service). Multi-dimensional (MD) recommender systems are used in order to incorporate more dimensions such as contextual information (the user, the product, the time, the weather... etc). All these dimensions can be useful for the process of decision making that subsequently results in recommendations.

Beside the attempts the researchers have done to apply MD and use the contextual information using a hybrid recommender system, we have decided to enhance this approach of MD recommender system [6]. This MD recommender system [6] was tested based on movies datasets from which the traditional two-dimensional CF approach was applied. Reduction based theory has been used in order to reduce the multi dimensions to two dimensions thus allowing to use contextual information as additional dimensions. This technique is elaborated more in the following section. However, to the best of our knowledge, there has not been any attempt so far to use this MD with hybrid recommender system (Knowledge-base) as its base and where multi-agents are deployed.

In this paper we propose a multi-agent recommender system that is based on multi-dimensional rating approach using the Knowledge-base Hybrid recommender system. The Knowledge-base which will be using rules in order to provide relevant recommendations to the users based on their profiles and contextual information. The system is also supported by a set of rules along with the contextual information. The recommendations are presented to the user in a pro-active fashion.

Following this introduction the paper is organized as follows. Section 2 presents an overview of the technology used for this work. Section 3 reviews significant research related work. Section 4 discusses the requirements of the design and then describes the system design architecture along with the inference engine of the system. Section 5 discusses the implementation of the system. Section 6 presents the system testing and evaluation. Finally section 7 concludes the paper and highlights the future work of this research.

¹ www.amazon.com

² www.netflix.com

II. OVERVIEW OF THE TECHNOLOGY USED

A. Recommender Systems

The internet is now recognized as the ultimate repository of information. However, this overabundance of information causes a problem to internet users when searching for specific information or services [7]. In order to solve this problem of filtering the information coming from the internet and to meet the user's needs so that those services are recommended frequently, the concept of Recommender System was introduced [8]. This kind of systems is capable of predicting users' preferences and provides products or/and services based on previously rated items [1].

According to Figure 1 recommender systems can be classified based on the recommendation approach used in [9], the algorithmic technique that is applied in the recommender system [10] or the personalized services provided by the recommender system [2]. As for the first classification, there are three types of recommender systems: the Collaborative Filtering (CF) where the popular items with higher ratings are recommended to other users with similar profiles, the Content-Based Filtering (CBF) where recommended items are similar to the items that the user has selected in the past, and hybrid where the first two types of recommender system are combined in order to avoid their limitations.

Recommender systems can also be classification according to [2] into: Service based on the preferences stated by the user explicitly [11], Services based on the location of the user (location-based) [12], and Services based on the user context (context-aware) [13].

The context-aware recommender system can be considered as the most suitable to be deployed in certain application areas such as tourism, education, and health. For example, in the tourism domain the contextual information is useful to be considered when providing a user with a set of recommendations based on time, season, and his/her exact location.

In order to handle multiple contextual information, some two-dimensional algorithms can be extended to support multiple dimensions as discussed in [6]. In that work the authors propose an approach using reduction based theory to implement a multi-dimensional recommender system.

The reduction-based approach is combined with a traditional two-dimensional estimation method (CF). The main advantage of using reduction-based is by reducing the problem of the multi-dimensional recommendations to the two-dimensional recommendation space so that after the reduction almost any two dimensional technique can be used [6]. For instance a recommendation R of a movie that the user is going to watch during the weekend can be represented as:

$$R_{User \times Content \times Time}^D (John, Action, Weekend) \quad (1)$$

where D is the set of existing ratings which contains records <user ,content, time, rating> for each of the user ratings. The overall set of ratings for the user John will first eliminate the time as the contextual additional information by selecting all weekend ratings for movies. After

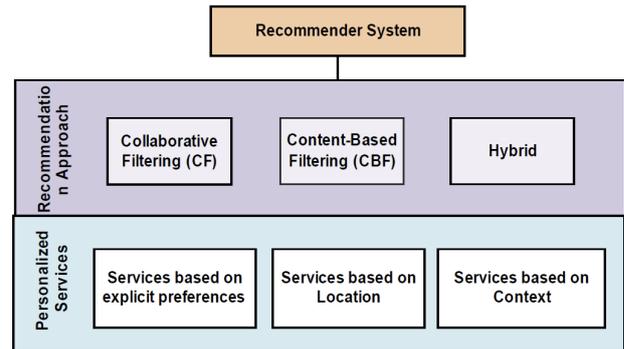


Figure 1. Classification of recommender system

that, it will become a traditional two-dimensional (User x Content).

III. RELATED WORK

In order to discuss the related work we emphasis on the issues of the traditional recommender systems as it was discussed in [1].

A. Issues of Content-based Filtering (CBF)

- **New user problem:** This depends on the user ratings. The less the number of ratings available by the user, the less accurate recommendations s/he will receive.
- **Overspecialization:** Only similar items to what the user has rated earlier will be recommended. However, users sometimes may want to change or try new things that they have not selected before.
- **Limited Content Analysis:** Similar features of two items might make them indistinguishable [14].

B. Issues of Collaborative Fieltering (CF)

- **New user problem:** The less number of ratings there are, the fewer amount of preferences are defined for a user and hence the recommender system provides less accurate recommendations.
- **New item problem:** The item which is rated the most by users will be highly recommended. Yet, it happened that there are new items and they are interesting but because they are not rated before, they will not be recommended to users.
- **Sparseness:** This refers to the number of people who rate an item compared to the ratings themselves. Some items are rated highly but very few people have made those ratings.

Felfernig et al. in [15] argue that collaborative filtering used by, for example, Amazon, cannot be used for tourism applications. The reason being is that collaborative filtering performs best when there are already significant numbers of rated items since the classifications of books for example are known. Moreover, some books are classic in a way that even though they were published 40 years ago, a group of people can still buy such books. As for tourism the personal trip planning is less frequent compared to buying a book. It is rarely the case that tourists will be interested in the exact same events. As it was stated earlier, that collaborative filtering particularly was not sufficiently used yet when it comes to the tourism domain [16]. Tourist mobile guides usually tend to provide a tourist with recommendations that are closer to his/her preferences, which allow more personalization of the suggested

recommendations. Yet, these guides fail to use what other tourists might like and highly rate to be recommended [2]. Additionally, the recommendations are normally fired based on an explicit request from the user. However, this is not suitable especially when sudden events occur like change of locations and activities. This can be applied for emergency situations or even when a tourist for instance is on a trip where specific festival or concert is running at the same time and place s/he is in. The sections below explore some solutions to find a suitable recommender system approach for the tourism domain.

C. Available Solutions

The available solutions of most of the issues mentioned earlier are:

1) Hybrid recommender system

A hybrid approach attempts to overcome the limitations of CF and CBF techniques by combining them to provide more accurate recommendations than those only based on CBF or CF [9, 17]. Based on how the two approaches are combined, one of the classifications of hybrid recommender systems is the unified model which has properties from both content-based and collaborative approaches. There are many models that have been designed and implemented under this category, which mainly uses methods from the content-based and the collaboration approaches with other techniques and methods. One of the techniques that have been used is the knowledge-based as in [18]. The knowledge-based technique improves the accuracy of the recommendations brought by hybrid recommender systems and deal with the limitations of recommender system approaches (cold start) [1]. The cold start concept refers to the problem the recommender system is facing when there is a new user or a new item as stated earlier [19]. The knowledge-based recommender systems are applied in the tourism domain which proved to have significant improvement. An example is the tourism recommender system [3].

2) Multi-agent recommender system

Choosing to utilize a multi-agent in a recommender system, especially in the tourism domain, came mainly from the need to facilitate the recommendation process to serve a larger scale of tourists with different services. Agents were used particularly to simulate the human intelligence by using a set of logic rules that each agent will have in order to perform certain tasks on behalf of the users [20]. An example of task-based recommender system is [5]. The tasks assigned to the agents are usually designed to solve smaller problems for more complex and larger problems [21]. Agents have the ability to be social and communicate with other agents and other components in the systems [22, 23]. The interaction between the agents can be done automatically in an intelligent way to pass the required information [24]. Moreover, agents are able to communicate any results directly to end user [25]. Agents are also able to monitor the users' behaviors in order to build their profiles, keep the history, and then take the feedback implicitly [26].

3) Multi-dimensional rating approach

Adomavicius et al. [27] proposed a multi-dimensional approach that allows using contextual information about the user such as time and location beside the basic two dimensions that is used by most of the recommender systems (user and item).

Recommender systems were used originally in the tourism domain to enhance the process of arranging a trip independently with most of the services required, such as places to visit, restaurants, hotels and flights bookings. The previous solutions can be combined in order to form a recommender system. The multi-dimensional approach uses the contextual information to provide more accurate recommendations to the users, which is suitable for the tourism domain. The base of the multi-dimensional approach can be hybrid recommender system to avoid the limitations of CF and CBF. The multi-dimensional approach can be developed to be more pro-active and has a more efficient inference engine through integrating it with multi-agent systems. The following section describes the proposed multi-dimensional multi-agent system architecture for a tourism application.

IV. SYSTEM ARCHITECTURE

A. Description of the system architecture

In order to realize the system, the architecture is divided into four layers as illustrated in Figure 2.

They are as follows:

1) Data Layer

This layer is responsible for storing the data such as the profile of the users and the information about the services that are brought from the web which is cached and kept up-to-date to avoid network-induced bottlenecks.

2) Agent Layer

This is the core component of the system that contains the inference engine and the environment of the agents. The main parts of this layer are the agents themselves which are created by the Agents Engine Factory. Each agent is specialized in performing one task or one type of the three services provided by the system. In the case of the tourist prototype system implemented in this paper the services are the request of a hotel, a restaurant and/or an event. Each agent's intelligence will be defined as a set of rules. The agents use the multi-dimensional inference engine (MDIE) component in order to store the ratings of different recommendations which will lead to anticipating recommendations to users as it will be explained in the next section. Finally, there are the Information Retrieval Manager and the Profile Manager components. The former manages the information retrieved in the data layer while the latter manages the profiles of the users which are stored in the data layer.

There are four main agents that the system creates for each user based on his/her requirements of services. Their functionalities are explained below:

- **Schedule Agent:** The schedule agent is created after the minimal information required is collected from the user. Each user will have his/her own schedule agent that will be responsible of creating a schedule that will be filled later with trip activities. The schedule agent will be in charge of sending requests to the agents' engine to create other agents when needed. Moreover, the schedule agent will also be in charge of sending reminders of the activities to the users. Once the user makes a selection, feedback will be taken implicitly and passed to the profile manager to build/update the profile of the user.
- **Restaurants Agent:** The restaurants agent is created by the agents' engine to bring recommendations of

restaurants to the user. This agent will first request the profile manager to bring specific information that is useful such as the type of trip and check if there is history or not. The information pulled from the database will then be passed to the MDIE entity to predict ratings of recommendations to be brought by the restaurants agent. The recommended restaurants will be passed to the information retrieval manager to store them in the retrieved information database so it can be used later on.

- **Hotels Agent:** the hotels agent will be created by the agents' engine before even the schedule agent fills the schedule with the activities. This agent will be created based on the need for a hotel or not. Based on the trip dates that the user would specify at the beginning, it will be decided whether this hotel agent should be triggered or not. For example, if the user has inserted the start date of the trip to be equal to the current date or after, there might not be a use for recommending a hotel since the user has already started the trip.
- **Events Agent:** the events agent will be created based on the need to provide events recommendations. It will access the profile and check if a history exists. The information found will be passed to the MDIE where the recommendation ratings will be predicted and provided to the user.

3) Control Layer

This layer facilitates the transition of the information from the user terminal to the agent's layer and vice versa.

4) View Layer

This layer provides a friendly user interface that a traveler would interact with through the system.

B. Multi Dimensional Inference Engine

In this section we will discuss how the inference engine works for one service which is the restaurants yet it works the same for the other services.

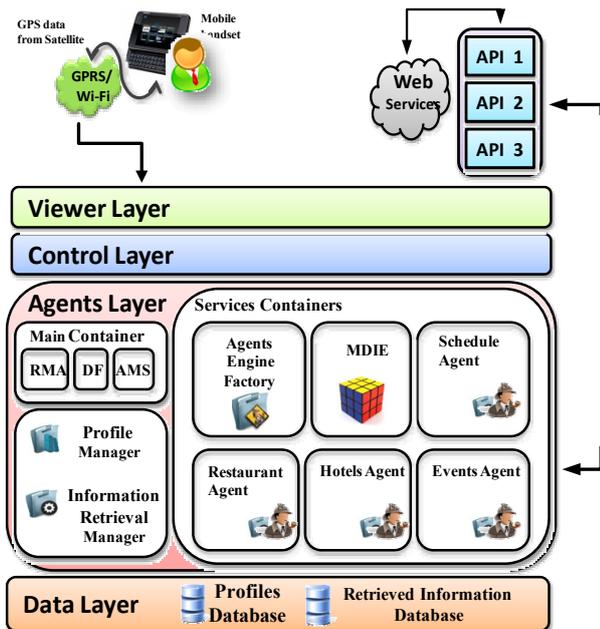


Figure 2. Design Architecture

1) Multi-dimensional recommendations cube

As discussed in [28] implicit feedback can be taken from the behaviors of the users, e.g.: selections, duration, and deletion. However, since the interaction between the user and the system will be reduced to the absolute minimum, in our case only the selections of the recommendations proposed to the user are considered. Therefore, the ratings of the recommendations are taken implicitly by observing the user's behavior and the selections a user makes. The ratings are going to be saved in the database. It can be represented in the recommendations cube as it is the case of the multi-dimensional approach [6] where Adomavicius et al. have followed the multi-dimensional data model of [29].

There are 6 sides for one cube. Each side is considered as a dimension and represented in tables. Each table contains other useful information too. In our proposed architecture we propose to have three different multi-dimensional cubes in a way that each cube is assigned to a service since we have three services provided by the system. There is a slight difference among the three cubes based on the service dimension. We are going to go through all cubes to show how the combination of the dimensions can be used to store ratings.

Figure 3 illustrates the cube of recommendations ratings for the restaurants service. We have decided to use four sides and assign four dimensions, one for each one of them. The dimensions are: user, restaurants, time, and trips. A rating from these dimensions can be represented as the following:

$$R : USER \times RESTAURANT \times TIME \times TRIPS \rightarrow Rating \quad (2)$$

For instance, as shown in the Figure 3, a user *Ahmed* has been on a family trip in which he stayed in Abu Dhabi. In a weekday he had a meal at an Italian restaurant also in Abu Dhabi. The rating that was given to this combination of these dimensions is 3 according to the user's previous selections and preferences.

2) Recommendation probability

Conditional probability was used since the expected properties of the different recommendations provided to users can be used with different contextual information. The previous ratings which are saved in the ratings cube are going to be the base for getting the preferences of the

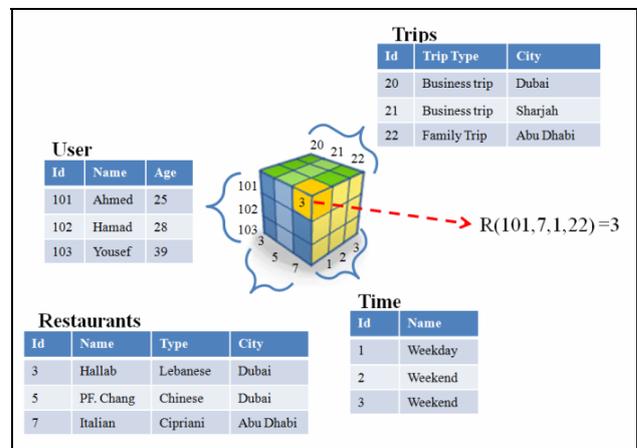


Figure 3. Multi-dimensional model for ratings of restaurant recommendations

users, and sorted according to their priorities. To explain this process we will demonstrate an example involving restaurants. The main attributes of the restaurants are the city and the type of food.

The ratings of the restaurants that a user has been to should be brought from the recommendation cube, as was explained previously. Table 1 shows all the existing ratings which were given by the system to the restaurants properties to the same user. By using a set of rules that will be explained later, along with the reduction based theory, the system would check the current location of the user, which for the example here is assumed to be *Abu Dhabi*. The first step is when the system would use the conditional probability to check the probability of which type of food the user would prefer on this visit, since the type of food is the first attribute.

Note that if the user has no such history random selection will be given to him.

The probabilities of each type of restaurants in Abu Dhabi are going to be calculated as the following:

$$P(\text{Type} / \text{AbuDhabi}) = \frac{A}{B} \quad (3)$$

where *A* is the rating given to specific type of restaurants in *Abu Dhabi* and *B* is the ratings that were given to restaurants in *Abu Dhabi* regardless of its type. Therefore, when calculating we have the following probabilities:

- P(Indian/Abu Dhabi) = 19/24 = 0.79
- P(International/ Abu Dhabi) = 2/24 = 0.083
- P(Lebanese / Abu Dhabi) = 3/24 = 0.125

According to Figure 4 Indian food has the highest probability that the user would like to have if was in Abu Dhabi. The first attribute that is the type of food will be assigned to be Indian food.

After this step we have the two attributes of the restaurants which are the city *Abu Dhabi* and the type of the food which is *Indian* as the highest priorities to the user. The process of selecting the attributes of hotels and places to visit will be applied in a similar way. The attributes of the hotels are: the number of stars and the city, while the attributes of the places to visit are: the category of place and the city.

In the following section we are going to apply these attributes of the same example on the available restaurants to see what recommendations to send to the users.

3) Multi-attribute theory

The multi-attribute theory is actually an evaluation methodology that is widely used [30] to assess and rank alternatives significantly [31]. Therefore, we have decided to use it in order to assess the recommendations before providing them to the users.

The attributes priorities that were shown previously are going to be used and applied using the multi-attribute theory. The multi-attribute choices are represented in Table 2 where each of the attributes will be assigned a weight according to the priority of the attributes (exact =2, was chosen before =1, never =0). The restaurants list would be brought from the web along with their attributes, food type and in which city they are located. For example, *Restaurant 1* serves Indian food and it is in Abu Dhabi. Therefore, the food type and the city are exactly matching user's preferences and thus both attributes were given the

TABLE I.
RATING OF RESTAURANTS THAT A USER HAS BEEN TO

| ID | City | Type of restaurants | Rating |
|----|-----------|---------------------|--------|
| 1 | Abu Dhabi | Indian | 15 |
| 2 | Abu Dhabi | Indian | 4 |
| 3 | Abu Dhabi | Lebanese | 3 |
| 4 | Abu Dhabi | International | 2 |
| 5 | Dubai | Indian | 5 |
| 6 | Dubai | Lebanese | 3 |
| 7 | Dubai | International | 2 |
| 8 | Dubai | International | 1 |
| 9 | Sharjah | Lebanese | 1 |

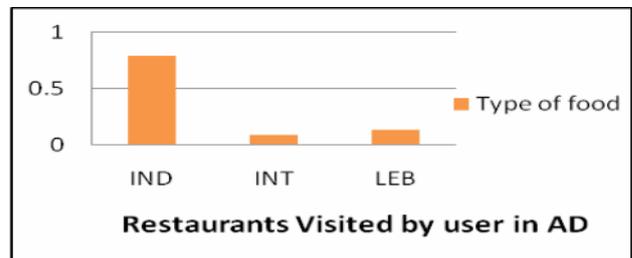


Figure 4. Probability of different types of restaurants

TABLE II.
APPLYING THE MULTI-ATTRIBUTE THEORY TO RECOMMENDED RESTAURANTS

| Recommended restaurants | Attributes | Food type | City | Total |
|-------------------------|--------------|-----------|------|-------|
| | Restaurant 1 | | 2 | 2 |
| Restaurant 2 | | 0 | 1 | 1 |
| Restaurant 3 | | 2 | 2 | 4 |

weight of 2. *Restaurant 2* serves Chinese food which the user has never experienced before so it would be given a weight = 0. Yet, city was for example Dubai that he has visited it, but it was not the highest when the probability was calculated. Weights for *Restaurant 3* are similar to Restaurant 1 since both are in Abu Dhabi and serves Indian food. Total of the weights are calculated then and the highest are going to be recommended to the users. In this example *Restaurant 1* and *Restaurant 3* are going to be recommended since they are the most suitable for the user.

The recommended services are going to be kept in the information retrieval database in case another user has common preferences and context can reuse these services or sometimes the same user would like to select another recommendation when making changes to his/her trip schedule

V. TESTING AND EVALUATION

The purpose of the system evaluation is to validate the system requirements according to the users' experience when using the system. Moreover, the system evaluation is conducted to basically measure the accuracy of the rec-

ommendations brought to users according to different profiles and contextual information. The system has been evaluated, also, against other similar systems.

A. Evaluation

The system was implemented and tested by real users as tourists in UAE. There are three different types of experiments that can be considered when evaluating a recommender system [19]: offline, online experience and user studies. The last type which is the user studies is what we have decided to use since it is the most suitable for our system at this stage since it is a prototype. The evaluation of the simulated prototype of the system was carried out by 71 volunteer who managed to do 220 test cases with different scenarios. They are from different educational backgrounds experiences, nationalities, ages, and genders. The system performed well and gave expected recommendations most of the time according to the evaluation conducted.

There are a number of criteria to evaluate a recommender system as in [33]. Based from the evaluation conducted we are going to check whether the system has satisfied these criteria.

- *Transparency*: The transparency is achieved partially by showing the users the recommendations as reminders and notifications prior the activity actual time. Any action taken is notified to the user so s/he can take it and hence update the schedule. The transparency can certainly be enhanced in the future by showing the user's previous selections and reasons of why the recommendations were suggested the first place. A number of users mentioned that in the comments.
- *Scrutability*: The scrutability is achieved by allowing the users to make changes to the trip schedule either manually or automatically. Manual schedule modification can be done by using the edit and delete buttons. On the other hand, the automatic schedule modification is done by the system by firing reminders and notifications. They allow the users to either accept or reject the recommendations. This what most of the users found it interesting about the system that it allows them to make the changes to the schedule either by themselves or by the system in case of rejecting a recommendation. Based on the activities in the schedule, the current location of the user will be taken and compared to where s/he should be. After that, the reminders will be sent to users. In case the location of different the system will sent the reminder as a notification of whether the user is willing to stick to the schedule or not. In case the user decided to not to stick to the schedule, new activities will be recommended to users in his/her current location. The schedule will be updated accordingly automatically. The selections the user made will also be taken into account since the ratings of the recommendations are taken implicitly.
- *Trust*: The trust is achieved in term of how the user is confident about the system starting with its recommendations and ending with how user friendly is the interface design. As shown in Figure 5 it turned out that 87% of the users prefer to receive recommendations based on their interests, 64% of them prefer to try new things very often, while 57% like to have same food types and visit places of their interest. The

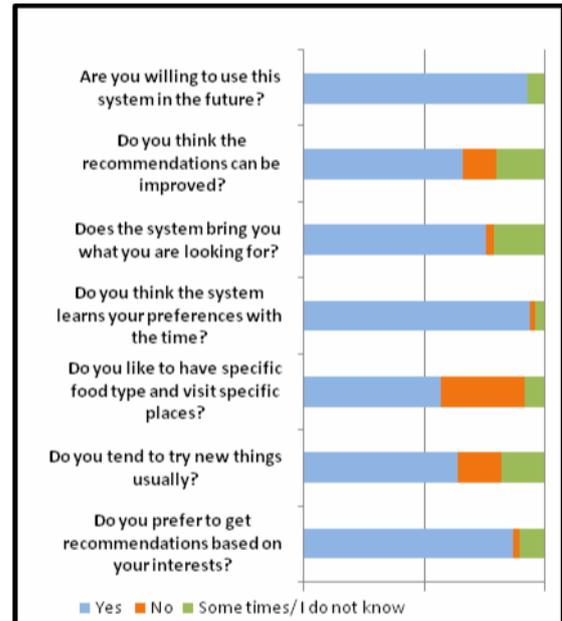


Figure 5. How users trust the system

majority of the users (94%) agreed that the system learns their preferences with time and 76% of them stated that the system brought to them the kind of recommendations that they were looking for while 21% found that it does some times. In spite of that, 66% of the users believe that recommendations can be improved and that other choices can be added to the system which is reasonable suggestions due to the limitations of the resources used for this prototype. It was a delight that 93% of the users showed that they are willing to use this system in the future, since they believed that such a system will be very helpful that they can trust when they plan for trips.

- *Effectiveness*: As for the system's prediction of recommendations for every user, it was found that 82% of the users have got what they have expected to find of recommendations which are similar to their preferences. Moreover, 93% of the users found that the system brings more accurate and better recommendations with time.
- *Persuasiveness*: this aspect is achieved by comparing the recommendations predicted with what users have in mind, and whether the user accepted the recommendations or not since in our system the users' ratings are taken implicitly and we cannot compare to actual users ratings. 80% of the users showed satisfaction about the recommendations while 17% of them ensured that it does some times. Moreover, 51% of the users rated the recommendations brought as excellent while 49% as medium which is also reasonable and fair since there is a number of users whom services ratings were showing a decrease and fluctuation.
- *Efficiency*: this is achieved by measuring the time the system takes in order to build a proposed plan full of restaurants and events/places recommendations. At the current stage with the limited number of resources we are using for the recommendations, the time they take to be calculated, checked, brought and filled within a schedule is almost 13 seconds at

maximum. The time is taken to bring a list of hotels that are according to users' preferences, is about 20 seconds at maximum.

- **Satisfaction:** 69% of the users found the system not demanding, and 87% agreed that the system is not noisy as well. Moreover, over half of the users (56%) believed that it requires less interaction than other systems. The other half found that it requires constant interaction all the time, or sometime. The reason behind this percentage is how the reminders and notification were designed to be fired in this prototype. They found that it was asking questions all the time. In the prototype the reminders of the schedule activities were designed to come one after the other because it is only a simulation. In real system a user might have one to two reminders/notifications in a day which cannot be considered noisy. Besides that, the system does not require attention from the user all the time, and it does not require a lot of steps in order to fire the recommendations. Furthermore, all of the users found the system easy to use and they only had to explore it for the first time only, and after that they did not take time to recall how to navigate through the system. Over 90% of the users found the interface design user-friendly indeed (Figure 6).

B. Comparison between available Systems

Since the architecture is supposed to maximize the pro-activity of a recommender system, there are a number of criteria that our system has met compared to the other projects as shown in Table 3. Creating a dynamic schedule trip is not provided by the other systems. Moreover, the contextual information is not always considered in most of the systems. On the other hand, the systems which use the contextual information mostly do not consider the accuracy of the recommendations that are provided to every user. The limitations of these systems give our proposed system the advantage of the pro-activity and portability of the system, and accuracy and dynamicity of the recommendations. The criteria are:

- **Least user effort/ interaction:** The complexity of any system can be measured by the number of steps a user has to go through in order to reach the result. The less the steps the easier the system to use.
- **Implicit feedback:** The feedback of the recommendations in any recommender system can be taken either implicitly by watching a user's behavior or explicitly by entering the feedback. Deciding to take the feedback implicitly affects the number of steps as described earlier.
- **Estimating Ratings:** The effectiveness of any recommender system can be measured by its ability to predict the ratings of items provided to users based on their preferences.
- **Using the contextual information:** Examples of the contextual information are location and time. Using such information can help providing more accurate and suitable recommendations to users.
- **Initiative:** The system can initiate actions and offer recommendations to the user and internally among the agents or the other components of the system.
- **Build/update profile implicitly:** This feature states that the user will have the ability to create his/her schedule. In case the user did not create his/her pro

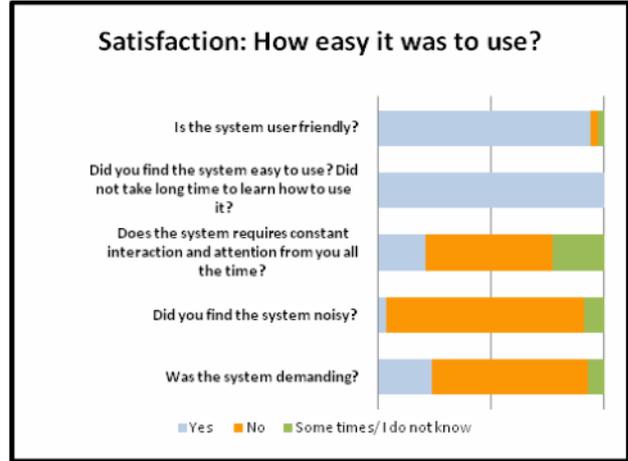


Figure 6. How users are satisfied

TABLE III. EVALUATION OF THE PROPOSED SYSTEM AGAINST OTHER SYSTEMS

| Criteria | Projects | | Multidimensional Recommender System [27] | Our Proposed System |
|---|--|-----|--|---------------------|
| | Hybrid Recommender System integrated Multi-Agent [4] | [5] | | |
| Least user effort /interaction | √ | √ | √ | √ |
| Implicate Feedback | x | x | x | √ |
| Estimating Ratings | √ | x | √ | √ |
| Using the contextual information | √ | x | √ | √ |
| Ability to handle changes initiatively | N/A | √ | x | √ |
| Pro-activity | √ | √ | x | √ |
| Build/update profile implicitly | x | x | x | √ |
| Usability of the system | √ | N/A | √ | √ |
| Accuracy of recommendations for every single user | √ | √ | √ | √ |
| Portability/ Heterogeneous | √ | √ | √ | √ |
| Create proposed trip schedule | √ | x | N/A | √ |
| Dynamicity of recommendations | √ | √ | √ | √ |

file, the system will do it. Also, updating the profiles will be done on behalf of the user.

- **Usability of the system:** The system usability measures how user-friendly the system is and how easy the user interface is.
- **Accuracy of recommendations for every single user:** This indicates whether the system is accurate when sending recommendations to users depending on each user's preferences and context.

- *Portability/Heterogeneity*: This measures whether the system works in heterogeneous environments
- *Create proposed trip schedule*: Some of the systems allow the users to create a trip schedule while others prepare this schedule automatically.
- *Dynamicity of recommendations*: The dynamicity of the recommendations indicates how they change according to user's profiles and context. They do not remain the same.

VI. CONCLUSION AND FUTURE WORK

Current recommender systems approaches have several limitations which include new user problem, overspecialization, limited content analysis, new item problem, and sparseness. Four types of hybrid recommender systems were introduced in order to overcome these limitations. Integrating the multi-agent system with the Knowledge-Based (KB) recommender systems has been proved to enhance its effectiveness. This is due to the agent technology ability to provide the pro-activity feature and the collaboration among the system components.

The main contribution of this paper is enhancing the multi-agent hybrid recommender system by applying the multi-dimensional approach so that it allows the system to consider more factors.

Architecture of a multi-agent recommender system that is based on multi-dimensional rating approach was proposed. It used KB hybrid recommender system as the base of the multi-dimensional rating since there are many benefits of this type especially when using rules for different cases. Such architecture can be applied to different scenarios, yet for this project it was applied in the tourism domain. The system will provide users with trip schedule within seconds. The system has the ability to build the profile of the user from scratch based on user selections. The contextual information is used to provide suitable recommendations of hotels, restaurants and events/places to visit. Agents are specialized in each of these services. The system behaves in a pro-active manner when providing the recommendations in a dynamic schedule. Thus, the interaction between the user and the system is reduced considerably. The feedback of the recommendations is taken implicitly from the user behaviour. The feedback is also used to update the profile of the user to store the preferences which will be used to provide more accurate recommendations in the future.

The system was evaluated by real users using the user studies as the evaluation experimental type. The evaluators performed 220 test cases in which different trips scenarios have been conducted. It was found that the system has satisfied the recommender system criteria.

The system was evaluated also against 3 of the related projects based on technical features like the implicit feedback and how heavy the interaction is.

As for future work, we are planning to maximize the system pro-activity by detecting the trips planned from the user's calendar. Currently since it is a web-based application it is not possible to access a device's calendar. However, if the system is implemented as a mobile application this can be done. Moreover, the mobile application can use the GPS system in any device to provide real time testing.

The inference engine of the system can be enhanced in order to understand the users type (which group) and hence bring more accurate and suitable recommendations. Furthermore, there are some services that can be added to the system to enhance the intelligence and help in bring more accurate and suitable recommendations such as: the weather, currency exchange, cinemas shows timing, and organized trips available with details and their prices.

Modeling different types of user behavior (such as irregular type) within the recommender system will satisfy the majority of users as currently the system only models regular behavior i.e. (the future looks like the past). This will require significant effort but will strengthen the system considerably.

The security of the communication between the server and the user could be improved so that the confidence of the users in the system can be increased. This will require the incorporation of some of the security strategy that is applicable to mobile agent systems.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734-749, 2005. <http://dx.doi.org/10.1109/TKDE.2005.99>
- [2] M. Kenteris, D. Gavalas, A. Mpitziopoulos, "A mobile tourism recommender system", IEEE Symposium on Computers and Communications (ISCC), Riccione, Italy, 22-25 June 2010, pp. 840-845.
- [3] T. Berka, and M. Plössnig, "Designing recommender systems for tourism", *Proceedings of The Eleventh International Conference on Information Technology in Travel & Tourism, ENTER 2004*, Cairo, Egypt, January 2004.
- [4] A. Casali, A. Von Furth, L. Godo, C. Sierra, A Tourism Recommender Agent: From theory to practice, WASI-CACIC 2007 (ISBN 978-950-656-109-3) pp. 1548-1561, Corrientes, Argentina, 2007.
- [5] F. Lorenzi, F. Correa, A. Bazzan, M. Abel, F. Ricci. A Multiagent Recommender System with Task-Based Agent Specialization. *Proceedings of the 10th Workshop on Agent-Mediated Electronic Commerce (AMEC-X)*. IFAAMAS, pp. 103-116, Estoril, Portugal, 2008.
- [6] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *JTOIS*, 23(1), 103-145, 2005.
- [7] P. Maes. Agents that reduce work and information overload. *Commun. ACM*, 37(7):30-40, 1994. <http://dx.doi.org/10.1145/176789.176792>
- [8] P. Resenick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings ACM Conference on Computer-Supported Cooperative Work*, North Carolina, USA, 1994, pp. 175-186.
- [9] M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Comm. ACM*, vol. 40, no. 3, pp. 66-72, 1997. <http://dx.doi.org/10.1145/245108.245124>
- [10] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence*, July 1998, pp. 43-52.
- [11] D. Kenteris M. Gavalas, D. Economou. An Innovative Mobile Electronic Tourist Guide Application. *Personal and Ubiquitous Computing*, 13(2):103-118, 2009. <http://dx.doi.org/10.1007/s00779-007-0191-y>
- [12] S. Berger, H. Lehmann, F. Lehner. Location-Based Services in the Tourist Industry. *Information Technology & Tourism* 5(4):243-256, 2003.
- [13] A. Hinz, G. Buchanan. Context-Awareness in Mobile Tourist Information Systems: Challenges for User Interaction. *Proceedings of the Workshop on Context in Mobile HCI*, Austria, 2005, pp. 257-266.

- [14] U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Month'," *Proc. Conf. Human Factors in Computing Systems*, 1995.
- [15] A. Felfernig, S. Gordea, D. Jannach, E. Teppan, M. Zanker: "A short Survey of Recommendation Technologies in Travel and Tourism", *OEGAI Journal* 25 (7), Oesterreichische Gesellschaft fuer Artificial Intelligence, pp. 17-22, 2007.
- [16] J. Herlocker, J. Konstan, L. Terveen, J. Riedl. "Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*", 22(1): 5-53, 2004. <http://dx.doi.org/10.1145/963770.963772>
- [17] M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, pp. 393-408, 1999. <http://dx.doi.org/10.1023/A:1006544522159>
- [18] R. Burke, "Knowledge-Based Recommender Systems," *Encyclopedia of Library and Information Systems*, A. Kent, ed., vol. 69, Supplement 32, Marcel Dekker, 2000, pp. 175-186.
- [19] G. Shani, A. Gunawardana, "Evaluating Recommender Systems", Microsoft Research Report MSR-TR-2009-159, Nov. 2009, <http://research.microsoft.com/apps/pubs/default.aspx?id=115396>, Retrieved 15, April, 2009.
- [20] W. Bing, W. Fei, Y. Chunming, "Personalized Recommendation System Based on Multi-Agent and Rough Set," *2nd International Conference on Education Technology and Computer (ICETC)*, vol. 4, pp. 303-307, Shanghai, China, July 2010. <http://dx.doi.org/10.1109/ICETC.2010.5529675>
- [21] V.N. Marivate, G. Ssali, and T. Marwala. "An intelligent multi-agent recommender system for human capacity building", *Proc. 14th IEEE Mediterranean Electrotechnical Conference*, pp.909-915, Ajaccio, France, 2008. <http://dx.doi.org/10.1109/MELCON.2008.4618553>
- [22] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2004.
- [23] I. Rudowsky. "Intelligent Agents". *Communications of the Association for Information Systems*, Vol.14, pp.275-290, 2004.
- [24] S. Macho, M. Torrens, and B. Faltings, "A multi-agent recommender system for planning meetings," *In Proc. Of the Agents'2000 workshop on Agent-based recommender systems (WARS'2000)*, Barcelona, 2000.
- [25] H. Zhu, X. Song, S. Zhang, Y. Gao, P. Zhang, "Personalized Recommendation Algorithm Research Based on Multi-Agent," *International Seminar on Business and Information Management 2008, (ISBIM'08)*, Wuhan, pp. 177-180, June 2009.
- [26] Y. Blanco-Fernandez, J.J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, B. Barrangans-Martinez, M. Lopez-Nores, J. Garcia-Duque, A. Fernandez-Vilas, R. P. Diaz-Redondo, "AVATAR: An Advanced Multi-agent Recommender System of Personalized TV Contents by Semantic Reasoning," *In Proceedings of the Fifth International Conference of Web Information Systems Engineering (WISE 2004), LNCS (3306)*, Brisbane, Australia, pp.415-421, 2004.
- [27] G. Adomavicius, A. Tuzhilin, "Extending Recommender Systems: A Multidimensional Approach", *IJCAI-01 Workshop on Intelligent Techniques for Web Personalization (ITWP'2001)*, Seattle, Washington, 2001, pp. 4-6.
- [28] C. G. Harrison, D. M. Chess, and A. Kershenbaum, "Mobile Agents: Are they a good idea?", Technical Report, IBM Research Division, T. J. Watson Research Center, 1995.
- [29] S. Chaudhuri, And U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65-74, 1997. <http://dx.doi.org/10.1145/248603.248616>
- [30] R. Schaefer. "Rules for using multi-attribute utility theory for estimating a user's interests", *In Online Proceedings of the 9th GI-Workshop: ABIS-Workshop 2001*, pages available online at http://www.kbs.uni-hannover.de/~henze/ABIS_Workshop2001/final/Schaefer_final.pdf, Dortmund, Germany, 2001.
- [31] J. S. Dyer, P. C. Fishburn, R. E. Steuer, J. Wallenius and S. Zionts. Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years. *Management Science* 38 645-654, 1992. <http://dx.doi.org/10.1287/mnsc.38.5.645>
- [32] R. Campbell Ranganathan. "An infrastructure for context-awareness based on first order logic", *Personal and Ubiquitous Computing*, Volum 7 Issue 6, pp. 353-364, December 2003.
- [33] N. Tintarev, J. Masthoff: A survey of explanations in recommender systems. In: *ICDE Workshop on Recommender Systems & Intelligent User Interface*, 2007.

AUTHORS

H. Al Tair received her B.Sc. in Information Technology majoring Software Engineering from College of Information Technology, UAE University in 2008. She received her M.Sc. degree in Computer Engineering from Khalifa University in 2011. Currently, she is working towards her Ph.D. degree at Khalifa University (e-mail: hend.altair@kustar.ac.ae).

M. J. Zemerly obtained his PhD from The University of Birmingham, UK, in 1989. Since then he worked at various universities in UK such as UCL, Warwick and Westminster. He then joined Khalifa University in year 2000 as an Associate Professor where he is currently the Computer Engineering Program Chair and chairs the eServices and Networks Research Group. His research interests are in ubiquitous computing, and context aware mobile systems. Dr Zemerly published more than 70 journal and conference papers as well as many book chapters. He is also the co-editor-in-chief of a journal and currently co-program chair of the ICITST conference series. Dr. Zemerly is a senior member of the IEEE (e-mail: [jama@kustar.ac.ae](mailto:jamal@kustar.ac.ae)).

M. A. Al Qutayri received the B.Eng degree from Concordia University, Montreal, QC, Canada, the M.Sc. degree from University of Manchester, U.K., and the Ph.D. degree from the University of Bath, U.K., all in electrical and electronic engineering in 1984, 1987, and 1992, respectively. From 1992 to 1996, he was a Senior Lecturer at De Montfort University, Leicester, U.K. He subsequently joined Khalifa University in the U.A.E., where he is currently a Professor in the Department of Electrical and Computer Engineering. He has published numerous technical papers in peer reviewed international journals and conferences in his fields of research, which include reconfigurable computing, embedded systems design and security, multi-agent systems and applications, and test and design for testability of mixed-signal integrated circuits. Dr. Al-Qutayri is a senior member of the IEEE, member of the IET, U.K., and holds the Chartered Engineer status from the U.K. Engineering Society. (e-mail: mqutayri@kustar.ac.ae).

Marcello Leida received an M.Sc. degree in Computer Science from the Università degli Studi di Milano, Italy and subsequently a Ph.D. in Computer Science from the Università degli Studi di Milano as part of a research collaboration programme between the Intelligent Systems Research Centre in BT and the Knowledge Management Group of Università degli Studi di Milano. During his Ph.D. he did research on Data Integration architectures, Ontology applications, Ontology Matching and uncertainty information management. He joined EBTIC in August 2009 and he is doing research on Enterprise Applications and Business Process analysis. His interests are mainly focused on semantic web, knowledge management and artificial intelligence; he published several papers in international conferences, workshop and journals on these topics. He has been program committee and referee for several conferences and member of IFIP W.G. 2.6 on database semantics. (e-mail: marcello.leida@kustar.ac.ae).

Received 09 March 2012. Published as resubmitted by the authors 24 June 2012.