M. Elammari, R. Abdleati, "Towards a General Architecture for Building Intelligent, Flexible, and Adaptable Recommender System Based on MAS Technology", The 11th International Arab Conference on Information Technology (ACIT), Benghazi, Libya, 2010.

Towards a General Architecture for Building Intelligent, Flexible, and Adaptable Recommender System Based on MAS Technology

MOHAMED ELAMMARI and RABEIA .N. ABDLEATI

Faculty of Information Technology, University of Garyounis, Benghazi, Libya
elammari@garyounis.edu rr1456@yahoo.com

ABSTRACT

Nowadays, recommender systems have been used to reduce information overload and to find the items that are of interest to the user. Many techniques have been proposed for providing recommendations to consumers or users. All currently available recommender techniques have strengths and weaknesses. Thus, numerous researcher studies have attempted to develop techniques that would overcome the various limitations of current recommender systems by combining existing techniques in different ways. On the other hand, we have found that many currently available recommender systems are still designed for some restricted domains. This paper presents our attempt to use agent technology to enhance recommender systems based on agent's property advantages with the goal to analyze and design a general architecture easily adaptable to several domains.

Keywords: *Recommender Systems (RS), general architecture, Multi agent system (MAS), User modeling.*

1. INTRODUCTION

The recommendation phenomenon is very common in our daily life, we receive verbal recommendations from our friends or family; these recommendations are insufficient if they are based on limited knowledge. Nowadays, recommender systems are emerging as a growing application and research field in several domains of computing research, from artificial intelligence to information systems. The available literature provides many recommender system definitions, as researchers and developers define base on their own perception or approach. One recent and simple definition of recommender systems is " Recommender systems use the opinions of members of a community to help individuals in that community identify the information or products most likely to be interesting to them or relevant to their needs" [11].

Many techniques for giving recommendations to consumers or users have been proposed, but all individual recommender techniques have strengths and weaknesses. Thus, researchers are focusing on finding techniques to overcome the various limitations of current recommender systems by combining various existing techniques in different ways [5] [2] .In addition, other technologies, such as agent technology, can be used as an aid to enhance recommender systems.

In this paper, we propose a general architecture based on multi-agent system (MAS) for building recommender systems. The designed architecture can be used as guideline for development of specific new recommender systems more efficiently, as they will not have to be built from scratch. Furthermore, we used High-Level and Intermediate Model (HLIM) methodology to analyze and design the MAS that will be supporting the presented architecture.

The paper is organized as follows: Section 2 presents a brief recommender systems and agent technology background. Section 3 reviews the related literature. Section 4 and 5 describe our contribution in recommender systems development by using MAS. Section 6 presents a case study to illustrate work proposed by Appling in one specific domain. Finally, Section 7 summarizes the conclusions of this paper.

2. BACKGROUND

2.1. RECOMMENDER SYSTEMS

a- Recommender system components

Each recommender system consists of three basic components [8] [11]:

- Items to be recommended: such as books, movies, music, courses, web pages. etc.
- Target consumer preference profile: this profile is created after user preferences are identified through various techniques; the process is also called user modeling.
- The recommender algorithm − also called recommender methods or techniques: this component is the mechanism that generates recommendations.

Most researchers in this field have been focusing on the third component and some classified recommender systems accordingly.

b- Recommender system approaches

There are many approaches (methods, approaches or techniques) in recommender systems field, each with its strengths and weaknesses; thus many researchers attempt to overcome the various limitations of current recommender systems by combining different approaches into hybrid systems.

The basic recommendation approaches are content-based filtering (CBF), collaborative filtering (CF), and knowledge-based approach. Content-based filtering approach generates recommendations based on the correlation between the items' content and user's preferences. In other words, these systems recommend items that are similar to previous user preferences [10]. Examples of content-based recommender systems are Syskill and Webert [13]. In contrast, collaborative filtering approach attempts to simulate the recommendation sharing collaboration between the users. Thus, it gathers data about consumers' preferences and matches it with other users in the system then recommends items to the consumer.

Most of collaborative systems apply the nearest neighbor model to compute the recommendations [8]. An example of Collaborative recommender systems is GroupLens [14]. Finally, the knowledge-based approach exploits its item domain knowledge base for generating recommendations to the user, combined with reasoning about what items meet user's requirements [16]. An example of knowledge-based recommender systems is Entree [3].

In addition to combining recommendation approaches, some other technologies, such as agent technology, can be used as an aid to enhance recommender systems.

2.2. AGENT TECHNOLOGY

Recently, Agent technology has received a great deal of attention in most computer science domains, due to its autonomy that enables; an agent to determine the problem solving criteria without constant guidance from the user , and then solve the problems even if given a vague and imprecise specification [7], Moreover agent technology incorporates structures that enable representing knowledge, achieving goals, interacting with the environments, and responding to unexpected changes that occur in its environment [12]. These structures could provide a significant advantage if used in the recommender systems' field, where they can make the recommender systems capable to take personal preferences into account, intelligently aggregate opinions and relationships from heterogeneous sources and data, make systems scalable, protect privacy, create open systems, and provide recommendations minimal user involvement [11]. Multi-agent systems are systems comprised of multiple interacting agents. They appear in a host of different application domains; for instance, industrial applications − process control and air traffic control, and commercial applications − information management and electronic commerce [17].

3. RELATED WORKS

There are many studies conducted in the area of recommendation systems. Some attempt to improve one of recommender system components at the expense of other components. Alternatively, they focus on the use of new methods to be integrates into recommender systems, without attempting to solve the existing problems or quality of recommendation in their works. For example, Chaptini [4] investigated the use of discrete choice models as a radically new technique for giving personalized recommendations. The work focused on user modeling, and wasn't intended to overcome the weaknesses of any recommender techniques. Others focused on the improvement in the recommendation algorithms; Sotomayor et. al [15] investigated integrating Singular Value Decomposition (SVD) with collaborative filtering approach to enhance CF approach by reducing the dimensionality of recommender systems databases. In addition,

Tran and Cohen [16] presented architecture for designing a hybrid recommender system for electronic commerce that combined collaborative filtering and the knowledge-based approach. Thus, it can be seen that the majority of studies were restricted on certain domains. For example, Marivate et. al [9] took MAS approach to solve the problem of recommending training courses to engineering professionals.

4. THE PROPOSED ARCHITECTURE

4.1. ARCHITECTURE OVERVIEW

The present study follows the approach based on multi-agent technology and switching hybrid method to design a general architecture for building recommender systems. Figure 1 illustrates the proposed architecture. This architecture consists of the following major components:

- Preference gathering
- Recommendation generator
- Item collector

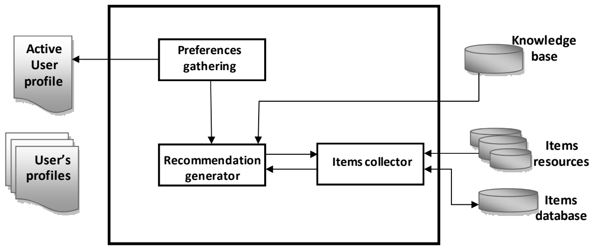

Figure 1: General architecture for building recommender systems.

4.2. ARCHITECTURE COMPONENTS

The generic functions of each component in the presented architecture are supported by agent technology. Each agent is responsible for a relatively simple task, but cooperatively they present the powerful recommendation.

a- Preferences Gathering Component

This component provides the main graphical user interface (GUI). It functions as an intermediary between the user and the system, and it responsible for interaction with the users to collect their preferences and display the recommendations. The graphical interface should select one of the following two subcomponents: (1**) profile generation and maintenance component** that will be activated when the user wants the system produce recommendation based on his/her preferences. This component is responsible for creating and updating an active user profile. It contains ***profiling agent (PA)*** that acts on behalf of the user to gather his/her preferences, as well as to build and update user profile.

Figure 2 illustrates the Profile generation and maintenance component.

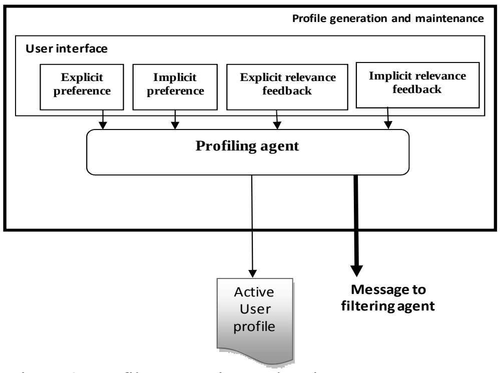

Figure 2: Profile generation and maintenance component.

After the user receives the recommendation, he/she can give feedback to the system. This feedback can be explicit, such as put rate(e.g. a system provide the user with list of items and ask he/she to provide his/her opinion via putting value to each item between 1 and 5), or implicit, such as purchase the product or put it in the basket.

The active user profile can be based on content analysis approach, as described by [1]. In this approach, the profile contains the information on the content of the items of interest. By applying this technique, the architecture can use CF and CBF as a hybrid approach.

(2) **User needs determination component** that will be activated when the user expects a recommendation based on his/her current needs. This component takes responsibility of gathering user's current needs. It comprises of ***need determination agent (NDA)***, which has capability to interact with the user and gather his/her requirements. After the user requirements are gathered, ***NDA*** transfers them to the ***filtering agent***. Figure 3 illustrates user needs determination component.

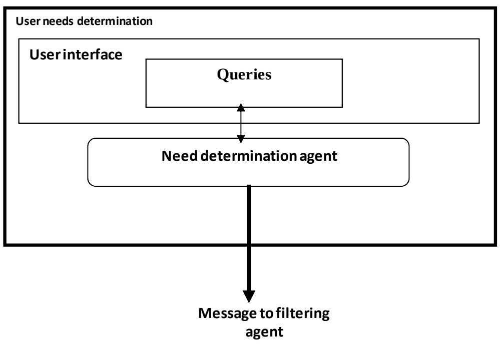

Figure 3: user needs determination component.

b- Recommendation Generator Component

The responsibility of this component (Fig. 4) is to generate the recommendations and translate them to the user interface. Within this component there is ***filtering agent (FA)***, responsible for producing recommendations via switching hybridization method. According to its up-to-date knowledge about active user, items, other users in the system and the situation (i.e. consider if the user is new or registered user in the system, if received message came from *NDA* or *PA*, etc...), it switches between the CF, CBF, and KB approaches.

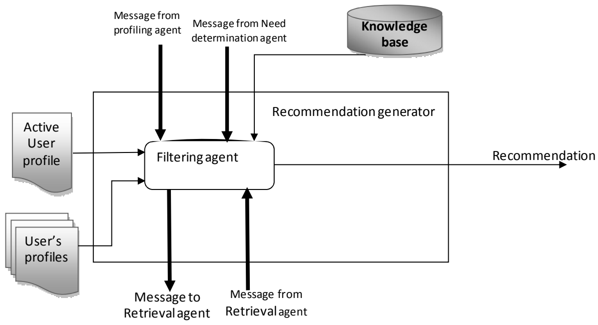

Figure 4: Recommendation generator component.

Filtering agent incorporates situation–action rules to switch between recommendation's approaches as illustrated in Figure 5. By following these rules, the presented architecture will avoid problems associated with: new system, new user, new item, unusual user, sparsity, and over specialization, all of which can be found in some existing recommender systems (see [2] for more details about recommender systems problems).

a. **If** received message from ***NDA*** or ***PA*** **Then**
send message to ***retrieval agent (RA)*** to request available items.
a. **if** the message came from ***NDA*** **Then** applies KB approach
Else /* if the message came from ***PA*** */
If new user **Then** apply KB approach
Else Compare available items list with active user profile and other users' profiles
If there is item found in available items list and not found in users' profiles
Then apply CBF approach.
Else
Apply CF approach.
a. **If** CF approach fails to produce recommendations **Then** switch to apply CBF approach.
b. Match active user profile with recommendation list, if there is item found in both **Then** remove it from recommendation list.
c. Match the recommendation list with available items, if there is item found in recommendation list and not found in available items list **Then** remove it from recommendation list.

Figure 5: Filtering agent action rules.

The **knowledge base** contains knowledge on how a specific item will satisfy the user's needs. The developer should have good understanding of the recommendation domain and should use appropriate methods for its development and updates, such as production rules or frames.

c- Item Collector Component

This component (Fig. 6) is responsible for collecting the items − and their features − related to recommendation domain, from the different resources. Within this component there is ***Retrieval agent*** that takes the responsibility item retrieval resource search.

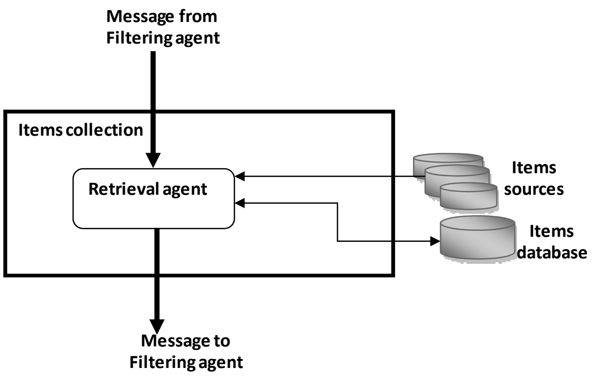

Figure 6: Item Collector Component.

The ***retrieval agent*** subsequently stores the available items and their features in the **item database**. This ensures that the system is updated with new items in the recommendation domain. ***Retrieval agent*** retrieves the items continuously in the offline stage (i.e. not during generating the recommendations), reducing the computation time. When the ***retrieval agent*** receives the message from ***filtering agent***, it extracts the items from the **items database** and sends the list of available items to the ***filtering agent***.

5. AGENT ANALYSIS AND DESIGN

For analyzing and designing the agents that support the proposed architecture, High-Level and Intermediate Models (HLIM) methodology, presented in [6], will be followed. HLIM models illustrate the external and internal system behavior, clarify the relationships between the agents in the system, and support the agent properties at design level. HLIM provides the system big picture by employing the use case map technique in the discovery phase.

5.1. THE DISCOVERY PHASE

In this phase, the use case map (UCM) technique is used to provide a high level view to the MAS that supported the proposed architecture. Figure 7 illustrates the high level view of the system.

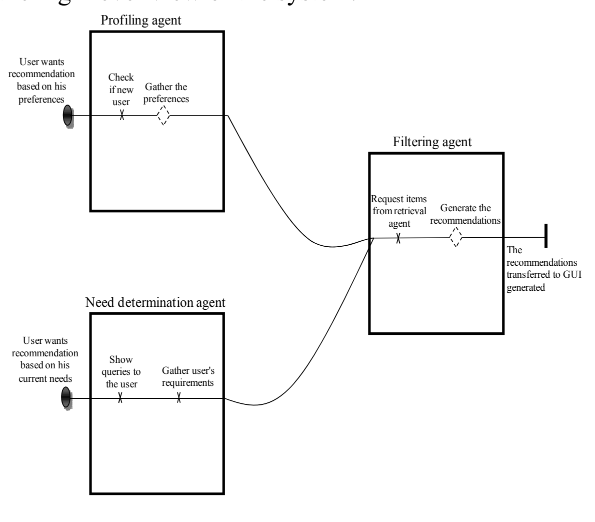

Figure 7: The use case map for the system.

Figure 8 illustrates Gather the preferences plug-in. figure (8-a) in case of new user, whilst Figure (8-b) presents the existing user case.

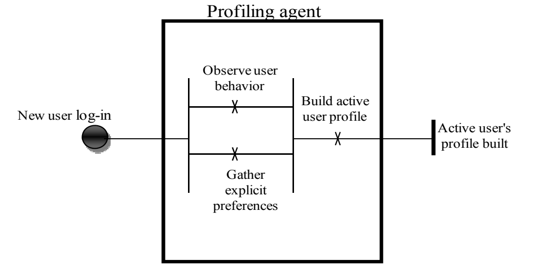

Figure 8-a: Gather the preferences plug-in. the case of new user.

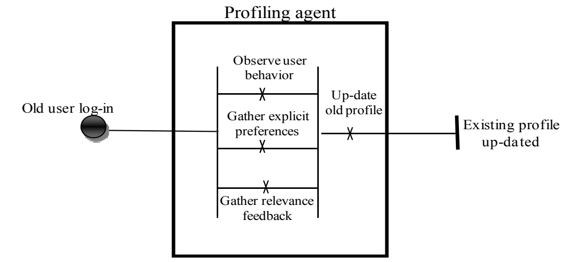

Figure 8-b: Gather the preferences plug-in. the case of existing user.

5.2. DEFINITION PHASE

This phase facilitates the MAS implementation that supported the proposed architecture.

a- Internal Agent Model

The internal agent model is used to illustrate the internal structure and behavior of each agent in the proposed architecture. Figure 9 illustrates internal model for ***filtering agent***.

filtering agent					
	Goal	**Precondition**	**Postcondition**	**Task**	**Comment**
1	Deliver recommendations to the GUI	Receive message from profiling agent or need determination agent	The recommendations transferred to GUI	• Request items from retrieval agent • Goal (Check the message resource) • Goal (Filter the recommendations list)	
2	Check the message resource	Message came from Need determination agent	The recommendations are generated	• Apply KB approach	Plug –in for check the message resource
3	Check the message resource	Message came from profiling agent	The recommendations are generated	• Check if new user apply KB approach • Check if there new item apply CBF approach else apply CF approach • Check if CF approach fail, apply CBF	Plug –in for check the message resource
4	Filter the recommendations list	Receive recommendation list	The recommendations transferred to GUI	• Compares active user profile with recommendation list • Remove the item that user has known before from the recommendation list • Transfer the recommendations to GUI	Plug –in for Filter the recommendations list
5	Filter the recommendations list	Receive recommendation list	The recommendations transferred to GUI	• Compares recommendation list with available items list • Remove the item that unavailable from the recommendation list • Transfer the recommendations to GUI	Plug –in for Filter the recommendations list

Figure 9: Filtering agent internal model.

b- The Agent Relationship Model

This model is used to clarify the relations between agents. Figure 9 illustrates relationship model for the agents in the proposed architecture.

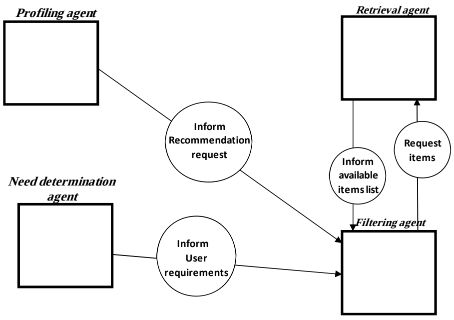

Figure 10: Agent relationship model.

c- The Conversational Model

This model is used to clarify the messages exchanged between the agents. The following abbreviations are used in the conversational models that related to proposed architecture:

PA: profiling agent
NDA: need determination agent
FA: filtering agent
RA: retrieval agent

Figure 11 illustrates conversational model for the filtering agent.

	Receive	Send	Comment
1	inform(from :PA to: :FA msg: there is recommendation request)	Request (from :FA to : :RA msg: give me list of available items)	
2	inform(from :NDA to: :FA msg: list of user requirements)	Request (from :FA to : :RA msg: give me list of available items)	
3	inform(from :RA to : :FA msg: list of available item)		

Figure 11: Filtering agent conversational model.

6. CASE STUDY

The following example shows applying the proposed architecture in one domain − a course recommender system.

It demonstrates how the proposed recommender system will be adapted to the structure of the proposed architecture.

It follows the steps outlined below:

- ***Student:*** Enter into the system
- ***System:*** Using graphical interface (Fig. 12), ask the student if he/she wants recommendations based on his/her preferences or not.

Figure 12: Determine what the user wants.

- ***Student:*** selected recommendations based on his/her preferences.
- ***System:*** Using graphical interface (Fig. 13), ask the student enter his/her username and password, or ask him/her to register in the system.

Figure 13: Determine if the student is new or registered user.

- ***Student:*** Enter as new user.
- ***System:*** Using graphical interface (Fig. 14), gather student preferences to build their profile.

Figure 14: Gather new student preferences.

- ***Student:*** Enter course names and browse courses list.
- ***System:*** Using graphical interface (Fig. 15), show the number of recommendations to the student.

Figure 15: Show recommendations to the student.

- ***Student:*** Submit his/her feedback on the recommendations to the system.
- ***System:*** Create and update the student profile.

7. CONCLUSION

This paper proposed general architecture that can serve as a guide for new developers in the field of recommender systems. The main architecture goal is to enable building specific recommender systems that can recommend items based on user preferences, gather user preferences with least involvement from the user, and recommend items in appropriate time, enabling the user to react immediately.

From the available literature, we could conclude that the presented work can be similar to Tran and Cohen [16] approach; Both used hybrid switched method and agent technology to enhance the recommendation and build architecture for building recommender systems. However, there are many differences between the two approaches; there are differences in the number of agents, number and nature of components and the switching strategy. Tran and Cohen [16] used one agent in the first component, this agent uses its knowledge about the number of users and number of rated items to switch between CF and KB recommendation techniques.

Each component in our architecture is supported by agent technology. The agent that support the third component responsible for switching between three recommendation techniques, which is CF, CBF and KB according to situation action rules to avoid the problems found in most existing recommender systems. Furthermore, we present analysis and design to the MAS that support the presented architecture while there is no work found in the available literature tailored to using agent's methodology, to analysis and design agent-based recommender system.

The developers using the designed architecture to build their bespoke recommender system, will benefit from the following:

- The architecture inherits CF, CBF and KB advantages.
- The architecture helps to avoid new system problem via KB approach.
- The architecture enables the recommender system to recommend new items to the user (solve new item problem) using CBF approach.
- The architecture enables the recommender system to produce cross-genre recommendation by applying CF approach.
- By using the content approach to build the user profile, the architecture helps to solve sparisty problem.
- The architecture gives the recommender system ability to recommend useful items to users who with unusual preferences (solve gray sheep problem) via CBF approach.
- Item retrieval from the different sources at offline stage reduces recommendation computation time.

There are further advantages in using multi-agent technology to design the presented architecture:

- The agent reactive ability gives RS enough flexibility to adapt to user's changing interests
- The proactive information gathering ability creates an up-to date RS whit new items in the recommendation domain.
- Recommender systems will be scalable, in view of the fact that they are inherently

modular, using MAS technology. Thus new agents can be easily added to the system when required.

- By providing an agent with a high-level goal to produce recommendation, it will act autonomously to achieve its goal.

REFERENCES

[1] Balabanovic M., Shoham Y., "Fab: Content-Based, Collaborative Recommendation" , *Communications of the ACM* , vol. 40, Pages 66 – 72, 1997.

[2] Burke R., "Hybrid Recommender Systems: Survey and Experiments", In *user modeling, and user-adapted interaction*, vol. 12, Pages 331-370, 2002.

[3] Burke R., *knowledge-based recommender systems*, In encyclopedia of library and information systems, vol. 69, 2000.

[4] Chaptini B., *Use of Discrete Choice Models with Recommender Systems*, PhD thesis, Massachusetts Institute of Technology, 2005.

[5] Cunningham P., Bergmann R., Schmitt S., Traphöner R., Breen S., Smyth B., "Websell: intelligent sales assistants for the world wide web", *e-business and e-work*, Venice, Italy, 2001.

[6] Elammari M., Lalonde W., "An Agent-Oriented Methodology: High-Level and Intermediate Models", *In 1st Int. Workshop. on Agent-Oriented Information Systems, Heidelberg, Germany*, 1999.

[7] Gill M., *Combining Mas and P2P Systems: The Agent Trees Multi-Agent System (ATMAS)*, PhD thesis, University of Stirling,2005.

[8] Itmazi J., *Flexible Learning Management System to Support Learning in The Traditional and Open Universities*, PhD thesis, University of Granada, 2005.

[9] Marivate V., Ssali G., Marwala T., "An Intelligent Multi-Agent Recommender System for Human Capacity Building", In *14th IEEE Mediterranean Electrotechnical Conference,* Ajaccio, France, pages 909-915, 2008.

[10] Montaner M., López B., De la J., "A Taxonomy of Recommender Agents on the Internet", *Artificial Intelligence Review, vol. 19, Pages. 285-330, 2003*

[11] Niinivaara O., *Agent-Based Recommender Systems*, Software Agent Technology Course Paper, University of Helsinki, 2004.

[12] O'Connor R., Gaffney E., *A Distributed Framework for Implementing a Multi-Agent Assistant System*, Technical Report CA1198, Dublin City University, 1998.

[13] Pazzani M., Muramatsu J., Billsus D., "Syskill & Webert: Identifying Interesting Web Sites", In proceedings of *the thirteenth national conference on artificial intelligence,* Portland, United States, vol. 1, pages 54-61, 1996.

[14] Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J., "Grouplens: an Open Architecture for Collaborative Filtering of Netnews", *ACM 1994 conference on computer supported cooperative work*, New York, NY, USA pages. 175–186, 1994.

[15] Sotomayor R., Carthy J., Dunnion J., *The Design and Implementation of an Intelligent Online Recommender System,* In proceedings of mixed-initiative problem-solving assistants, fall symposium series AAAI, 2005.

[16] Tran T., Cohen R., *Hybrid Recommender Systems for Electronic Commerce*, In knowledge-based electronic markets, papers from AAAI workshop, Technical Report WS-00-04, 2000.

[17] Wooldridge M., *An Introduction To Multiagent Systems,* john wiley & sons ltd, London, United Kingdom, 0-471-49691-x, 2002.