

An Intelligent Electronic Academic Advisor

Khamisa A. Yousef
Faculty of Science
University of Tobruk/Libya

Rabeia N. Abdleati
Faculty of Education
University of Benghazi/Libya

ABSTRACT

Academic advising is one of the important components in higher education. Advising can help students solve the academic problems they face during their study. In most world universities students select the courses that are most likely to meet their interests; therefore they need advising in choosing courses to complete their academic degrees. Academic advising is an essential element for an efficient learning process. This paper introduces an Intelligent Electronic academic advisor that adopts multi-agents technology using a recommendation system that switches between three recommendation techniques CF, CBF and KB, which helps students to decide which courses to take and presents academic advising for them.

Keywords

Academic Advising, Recommender System (RS), Multi agent system (MAS)

1. INTRODUCTION

The National Academic Advising Association (NACADA) defines academic advising as “a series of intentional interactions with a curriculum, pedagogy, and a set of student learning outcomes. Academic advising synthesizes and contextualizes students’ educational experiences within the frameworks of their aspirations, abilities and lives to extend learning beyond campus boundaries and timeframes” [1]. Effective academic advising will lead students to graduation in a timely manner, while poor advising or no academic

advising can cause problems with the course registration, progress, and possibly result in delayed graduation [2][3]. The quality of academic advising is also affected by the ratio of students per advisor, where a large ratio per advisor will cause deficient advice, unsuitable for course registration [3]. An Intelligent Electronic academic advisor is important, as well as a course registration process, to ensure the student achieves the university degree requirements in a comprehensive way through a structured approach and without facing unnecessary delays. This paper merges between MAS and a recommendation system to fulfill an intelligent system that helps students in advising procedures as an intelligent electronic academic advisor.

2. BACKGROUND

2.1 Multi-Agent Systems

A new way of analyzing, designing, and implementing complex adaptive software systems is Multi-Agent Systems [4]. Multi-agent system can be seen as “loosely coupled network of agents that work together as a society aiming at solving problems that would generally be beyond the reach of any individual agent” [5]. There are many benefits of using

multi-agent system for developing software systems, such as those given in [6], [7], [8] and briefly summarized below:

- MAS make systems scalable, in view of the fact that they are inherently modular, so it could be easy to add new agents into the system when required.

- MAS modularity could lead to simpler programming.
- Using multiple agents in the system could increase system’s operation speed by enabling parallel computation.
- MAS should make system development, integration, and maintenance easier and less costly, because the agents are comparatively independent software components that interact with each other via message-based communication systems.
- MAS efficiently retrieve, filter, and coordinate information from distributed sources.
- Due to dimensions of reliability, scalability, efficiency of computational, flexibility and maintainability, MAS enhances overall system performance.

2.2 Recommender Systems

2.2.1 Recommender System Components

Each recommender system consists of three basic components [9] [10]:

- 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.

2.2.2 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 [11]. Examples of content-based recommender systems are Syskill and Webert [12]. 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

[9]. An example of Collaborative recommender systems is GroupLens [13]. 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 [14]. An example of knowledge-based recommender systems is Entree [15]. In addition to combining recommendation approaches, some other technologies, such as agent technology, can be used as an aid to enhance recommender systems.

3. RELATED WORKS

There have been many studies conducted in the area of academic advising. Lawrence Keston Henderson and Wayne Goodridge [16] provided AdviseMe: An Intelligent Web-Based Application for Academic Advising that provides a reliable, user-friendly interface for the handling of general advisory cases in special degree programs offered by the Faculty of Science and Technology (FST) at the University of the West Indies (UWI). The system's core features include course advising, as well as information on graduation status and oral exam qualifications.

Fuhua Lin et al. [17] developed a system called e-Advisor. A Multi-agent System for Academic Advising was tailored to advise students in a single MSc Program. E-Advisor's functionality is dependent on very instructor in the department for the proper working of the system. While this extended staff involvement poses benefits in the context of Athabasca University, it fails to alleviate the staffing issues faced.

Marivate et al. [18] took a MAS approach to solve the problem of recommending training courses to engineering professionals. The recommendation system was built as a proof of concept and is limited to the electrical and mechanical engineering disciplines. The limitation of this system is in the use of the supervised learning neural network for ranking. Thus if a new discipline was added, the neural network would need to be retrained with new survey data.

Based on the brief summary of related work, it is evident that previous studies on Electronic academic advising system for academic advising provided useful information and some

innovative techniques. However, given the paucity of practical innovations in this field, in this work, MAS and the recommendation system are applied to build an Intelligent Electronic academic advisor, as a direct response to the problems the students are currently facing.

4. THE PROPOSED WORK

4.1 ARCHITECTURE OVERVIEW FOR ELECTRONIC ACADEMIC ADVISOR

The present study follows the approach based on multi-agent technology and switching hybrid method to design architecture for building an Intelligent Electronic academic advisor. Figure 1 illustrates the proposed architecture. This architecture consists of the following major components:

- Preference gathering
- Recommendation generator
- Item collector

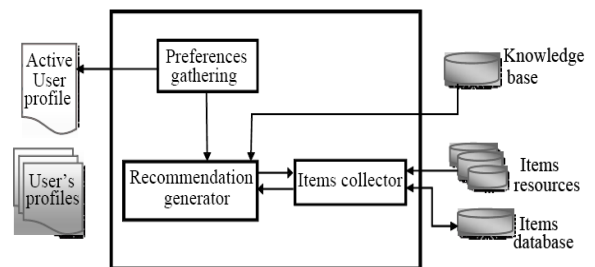


Figure 1: architecture for building Intelligent Electronic academic advisor.

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 advising. Fig. 2 shows the conceptual overview of the presented architecture to illustrate its component configuration and interactions in detail.

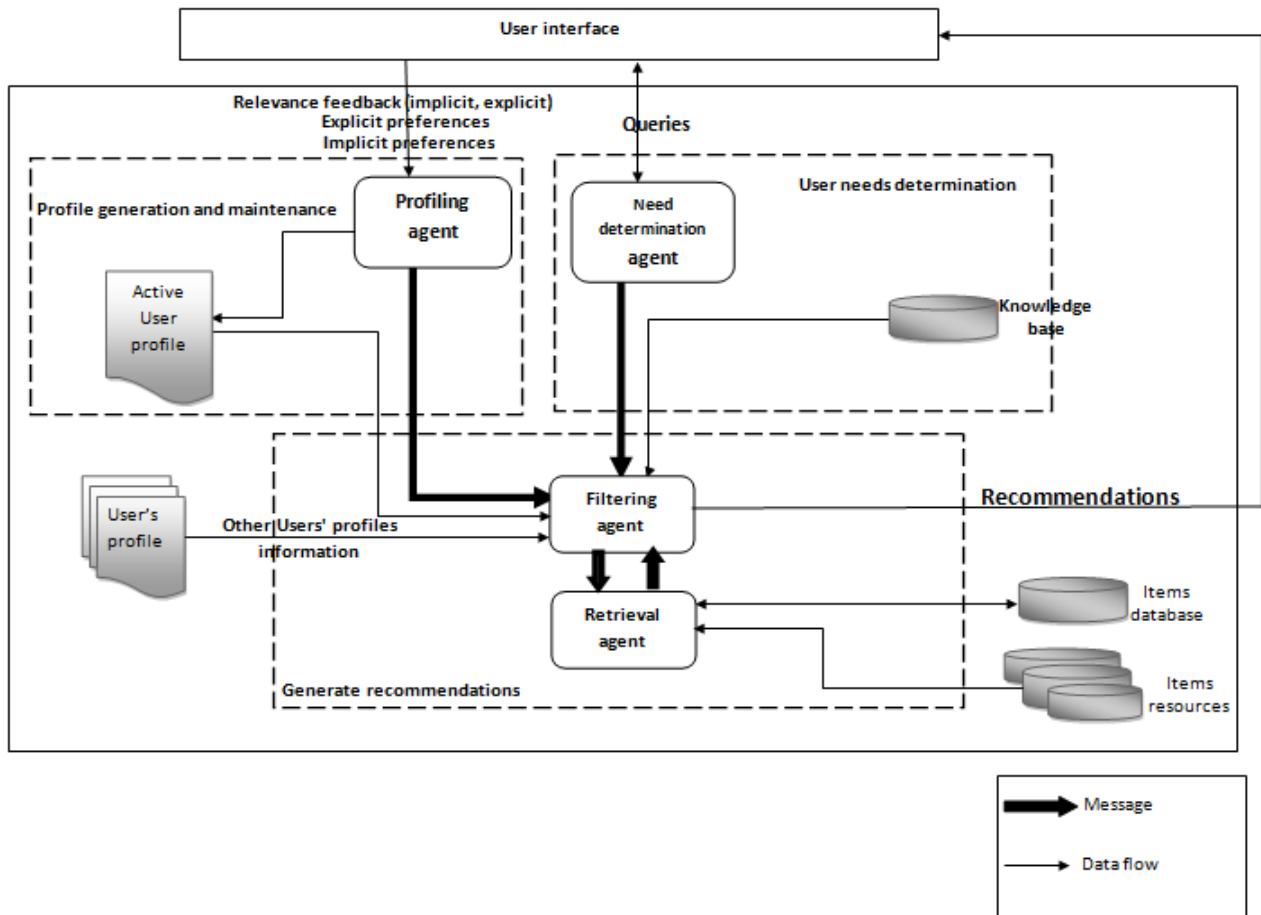


Fig. 2. Conceptual overview of the proposed Electronic academic advisor

4.2 Architecture Components

4.2.1 Preferences Gathering Component

This component provides the main graphical user interface (GUI). It functions as an intermediary between the student and the system, and is responsible for interaction with the students to collect their preferences and display the advising. Gathering user preferences is the first step in making the advising. If the system gathers sufficient information which enables creation of precise knowledge of student needs and preferences, the recommendations will be a closer match to the student requirements. The system can make the enquiry via the GUI, shown in Fig. 3.



Fig. 3. Determine student requirements.

The graphical interface should select one of the following two subcomponents:

4.2.1.1 Profile Generation And Maintenance Component

This component (Fig. 4) is responsible for creating and updating an active user profile. It contains a profiling agent (PA) that acts on behalf of the student to gather his/her preferences, as well as to build and update student profile. The profiling agent can explicitly ask the student about his/her preferences, or ask the student to rate a number of items that he/she knew before.

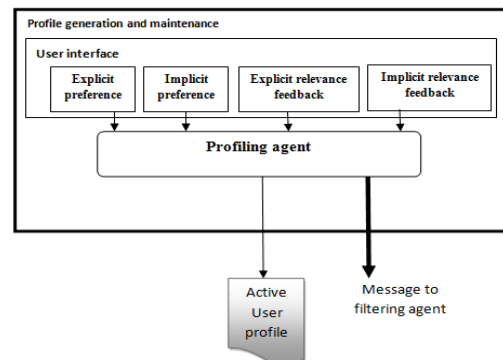


Fig. 4. Profile generation and maintenance component.

If the student is a registered student, the profiling agent use the gathered information to add new preferences to the existing student profile and delete preferences that had

changed; in this way the agent adapts to changes in student preferences over the time. The profiling agent transfers active student preferences to the filtering agent to inform it that there is a new advising request.

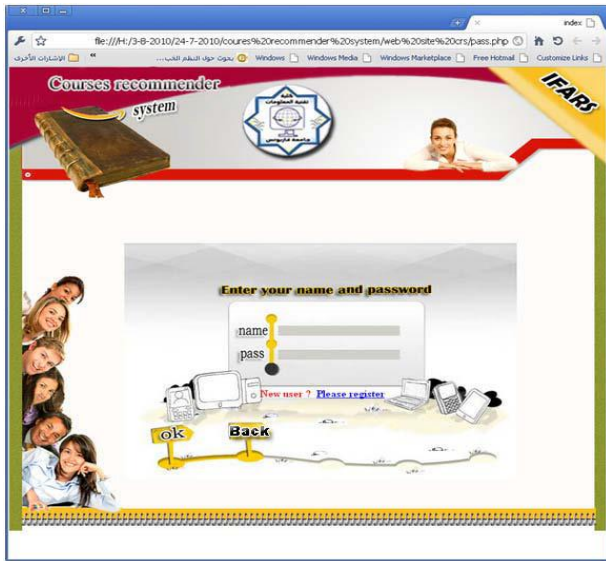


Fig. 5. Determine if the student is new or a registered student.

After the student receives the recommendation, he/she can give feedback to the system. This feedback can be explicit, such as rating the items (e.g. a system can provide the student with list of items and request his/her opinion by rating each

item on the scale from 1 to 5), or implicit, such as purchasing the product or placing it in a purchases basket. Profiling agent uses suitable implicit feedback methods to learn student preferences and update student profile. The following steps illustrate a simplified algorithm applied by the profiling agent:

- Step 1. The profiling agent interacts with the active student and checks if the student is an existing or a new user in the system.
- Step 2. In the case of a new student, the agent will observe the student behavior and gather his/her preferences, and then it will build an active user profile.
- Else /* In case of existing student */
- The agent will observe the student behavior, gather his/her preferences and gather his/her relevance feedback then it will add new preferences to the student profile and delete student preferences that had changed. In this way the agent updates active student profile.
- Step 3. The profiling agent informs the filtering agent of a new recommendation request.
- Step 4. The Profiling Agent waits for another interaction. If an interaction is received, it returns to step 1.

Figure 6 illustrates how the profiling agent works.

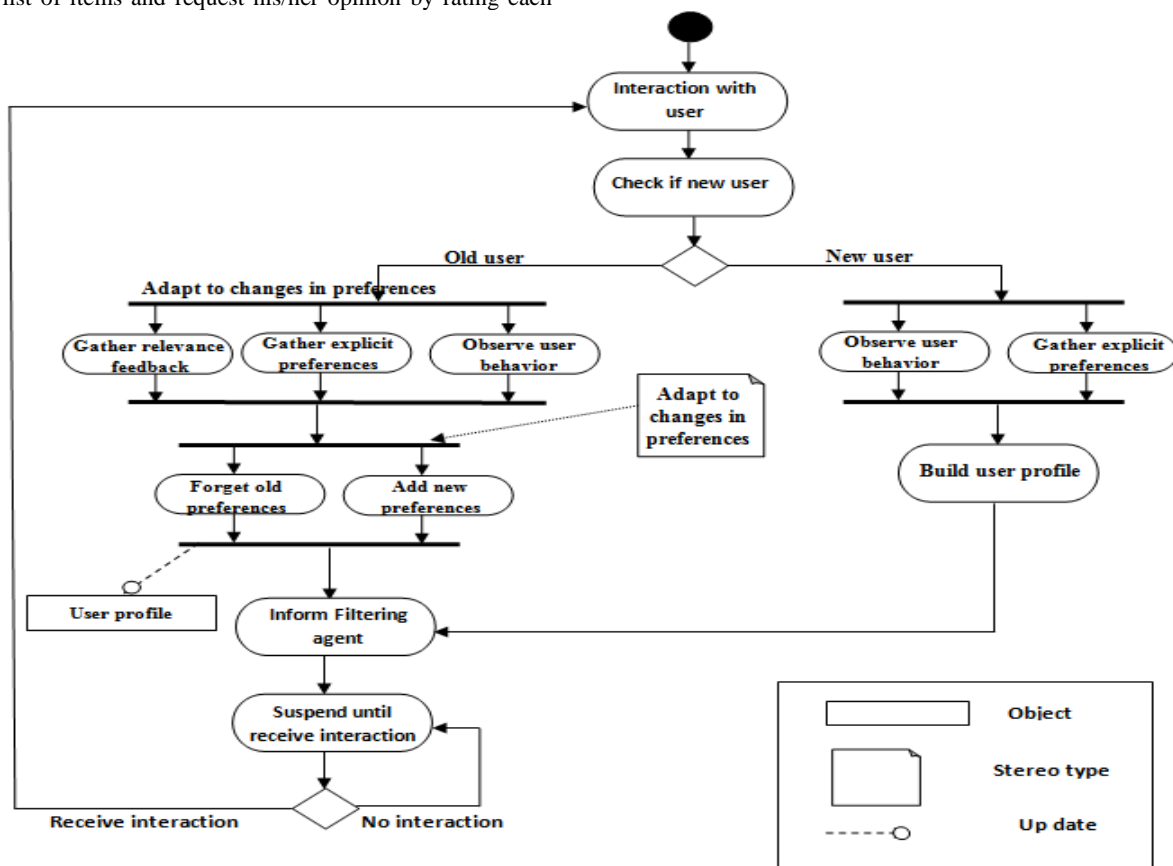


Fig. 6. Profiling agent algorithm steps.

4.2.1.2 User Needs Determination Component

This component (Fig. 7) takes responsibility of gathering user's current needs. It comprises of a *need determination agent (NDA)*, which has capability to interact with the user and gather his/her, requirements. The *NDA* can do that via queries (the developer chooses an appropriate way to show the queries, either as lists or questions). After the user requirements are gathered, the *NDA* transfers them to the *filtering agent*.

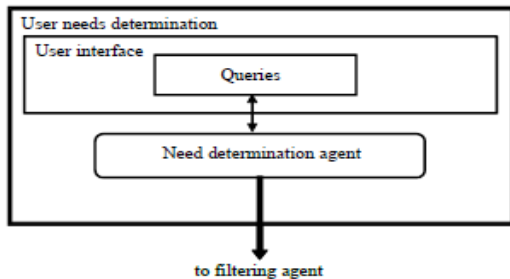


Fig. 7 User needs determination component.

The following steps illustrate a simplified algorithm applied by the need determination agent:

- Step 1. The need determination agent interacts with the student.
- Step 2. The agent gathers the student requirements.
-
- Step 3. The agent sends the gathered information to the filtering agent.
- Step 4. The agent is suspended until new interaction is received.

Figure 8 illustrates how the *need determination agent* works.

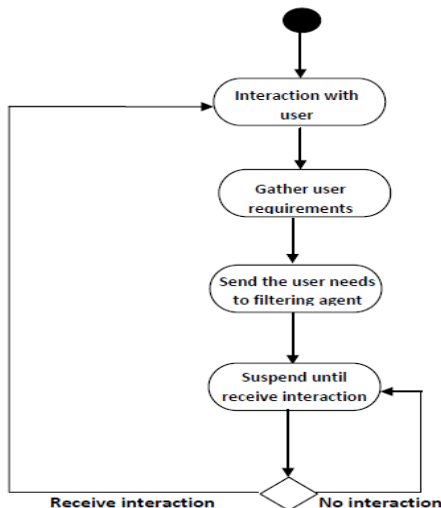


Fig. 8. Need determination agent algorithm steps.

4.2.2 Recommendation Generator Component

The responsibility of this component is to generate the recommendations and translate them to the user interface. Figure 9 illustrates Recommendation Generator component. Within this component, there is a filtering agent (FA), responsible for producing recommendations by applying a switching hybridization method. According to its up-to-date knowledge about the active user, items, other users in the

system, and the context (i.e. it considers whether the user is a new or a registered user in the system, if received message came from *NDA* or *PA*, etc.), it switches between the *CF*, *CBF* and *KB* approaches

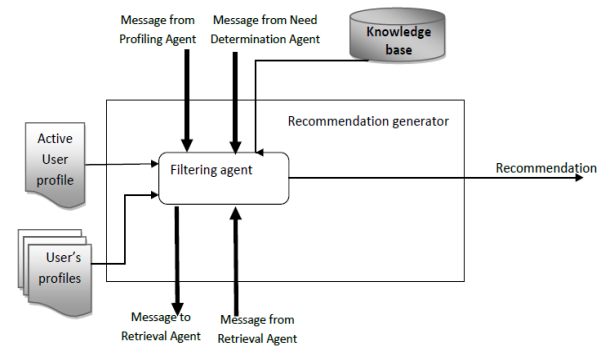


Fig. 9. Recommendation generator component..

4.2.3 Items Collector Component

This component (Fig. 10) is responsible for collecting the items _ and their features _ related to recommendation domain, from the different resources. Within this component, there is a *Retrieval agent* that takes the responsibility for searches in the resources to retrieve the items. These resources may be websites or internal database or both, according to the nature of the recommender system.

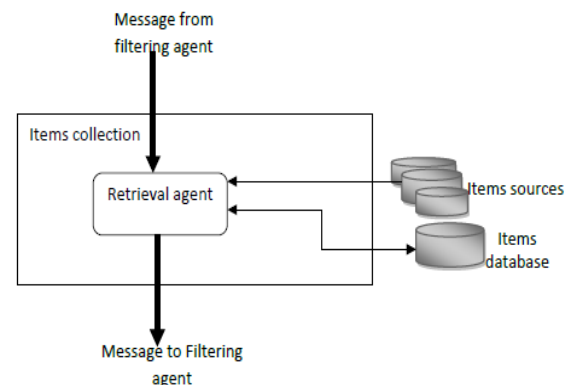


Fig. 10. Items Collector Component.

The *retrieval agent* subsequently stores the available items and their features in the *items database*. This ensures that the system is updated with new items in the recommendation domain. The *retrieval agent* retrieves the items continuously at offline stage (i.e. not during generating the recommendations); in this way, the recommendation computation time will be reduced. When the *retrieval agent* receives the message from the *filtering agent*, it extracts the items from the *items database* and sends list of available items to the *filtering agent*. The *retrieval agent* should be mobile agent. There are several benefits to using a mobile agent, such as reduction in the communication bandwidth, load balancing and overcoming latency of the network [19]. The following steps illustrate a simplified algorithm applied by the *retrieval agent*:

- Step 1. Search in the available recourses to collect available items.
- Step 2. Store the items and their features in the items database.
- Step 3. If a message is received from FA Then prepare the available items list.
- Else
- Go to step 1.
- Step 4. Transfer the available items list to FA then go to step 1.

Figure 11 illustrates how the *retrieval agent* works.

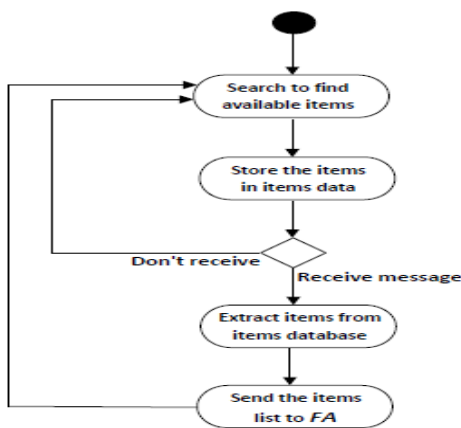


Fig. 11. Retrieval agent algorithm steps.

Case Study

This case study consist the use of scenario-based prototype to build An Intelligent Electronic academic advisor for course selection is shown, to demonstrate how the proposed Electronic academic advisor system will be adapted to the structure of the implemented architecture.

This case study provides scenario-based prototype for An Intelligent Electronic academic advisor, which helps students to decide which courses to take. The system can switch between three recommendation approaches (CF, CBF and KB) for advising courses to students. The recommended courses can be similar to other students' preferred courses, or similar to the courses that the student has enrolled before. Furthermore, the suggested courses can be a result of the extraction of the system knowledge domain, which recommends courses that the student may have interest in.

A number of characteristics for the course advising system in this case study are summarized below:

- User profile: each student registered in the system has a profile that contains his/her preferences and all the information related to the courses to which the student has enrolled in the past.
- Course resources: this case study assumes that the resource is an internal database that contains the courses that the university is offering. Additional information about these courses; such as their description and availability is also included.
- Domain knowledge base: It contains knowledge about how the specific course meets the student needs and it should be represented by rules or ontology.

Scenario1: a student wants advising based on his current needs; entered into the system. In this scenario, the NDA will interact with the student and give the student a range of questions to be answered (the agent can interact with the student via dialogues). The scenario when student wants advising based on his/her current needs can be summarized as:

- System: Please select suitable course category from the following: Information Systems, Software Engineering, or Computer Networks.
- Student: selects Software Engineering
- System: Please select courses you like from the following: Foundation of software engineering, Software project management, Large scale software design, or Re-use and component based development.
- Student: selects the courses he/she is interested in.
- After the student has answered all the given questions, the NDA will then send student requirements to the FA as illustrated in the figure 12.

(Inform	
: Sender	need determination agent
:receiver	filtering agent
:language	AgentSpeak
:ontology	Courses recommendations
:content	student requirements

Fig 12. Message from the need determination agent to the filtering agent

The agents within the system can exchange their messages via one of agent communication languages. In this case study, the exchanged messages are illustrated using KQML. According to [72], KQML communication language is a high-level, message-oriented information exchange protocol. KQML is independent of applicable ontology, transport mechanism (such as TCP/IP, SMTP, etc.), content language (such as Prolog, AgentSpeak, etc). Furthermore, KQML is independent of the ontology assumed by the content.

When the filtering agent receives the need determination agent's message; it will send message to the retrieval agent to request the available courses as illustrated in Figure 13

(Request	
: Sender	filtering agent
:receiver	retrieval agent
:language	AgentSpeak
:ontology	Courses recommendations
:content	Give me available courses

Fig 13. Message from the filtering agent to the retrieval agent

After the retrieval agent receives the message, it will search in the courses database and retrieve available courses then send the content to the filtering agent. The retrieval agent message is illustrated in Figure 14.

(Inform	
: Sender	retrieval agent
:receiver	filtering agent
:language	AgentSpeak
:ontology	Courses recommendations
:content	List of available courses

Fig 14. Message from the retrieval agent to the filtering agent

After computing the recommendations, the filtering agent will filter the content and then display the recommendations to the student. The system can show the advices via the GUI, as example of which is given in Figure 15.



Fig 15. Display the advices to the student.

Scenario 2: a new student requests advising based on his/her preferences entered into the system. In this scenario, initially the system has no knowledge about the student, and the PA will create a new student profile and inform the filtering agent to give the recommendations to the student. This scenario can be summarized as:

- Student: enter as a new user.
- System: gather student preferences via GUI to build his/her profile. An example of the new student GUI display is shown in figure 16.

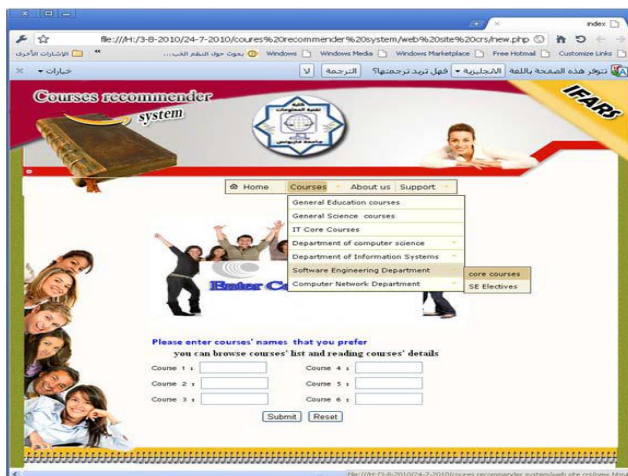


Fig 16. Gather new student preferences

After the student enters the preferred course names, and browses the available courses details; the profiling agent will build a profile for the student that contains student preferences. It will then send a message to the filtering agent informing it that there is a new recommendation request. The profiling agent message is illustrated in figure 17.

(Inform	
: Sender	profiling agent
:receiver	filtering agent
:language	AgentSpeak
:ontology	Courses recommendations
:content	There is recommendation request

Fig 17. Message from the profiling agent to the filtering agent.

After computing the recommendations, the filtering agent will filter the content and display the results to the student. The system can show the recommendations via GUI, as depicted in Figure 12. When the student submits his/her feedback about the recommendations to the system; the profiling agent will update the student profile.

Scenario 3: a registered student requests the course recommendations based on his/her preferences entered into the system. The scenario steps in this case can be summarized as:

- Student: enters the username and the password.
- System: asks the student to enter additional courses he/she prefers and monitors the student behavior within the page, including observing which courses he/she is browsing to gather student's preferences and feedback for updating his/her profile. The system achieves this task via GUI, an example of which is shown in Figure 18.

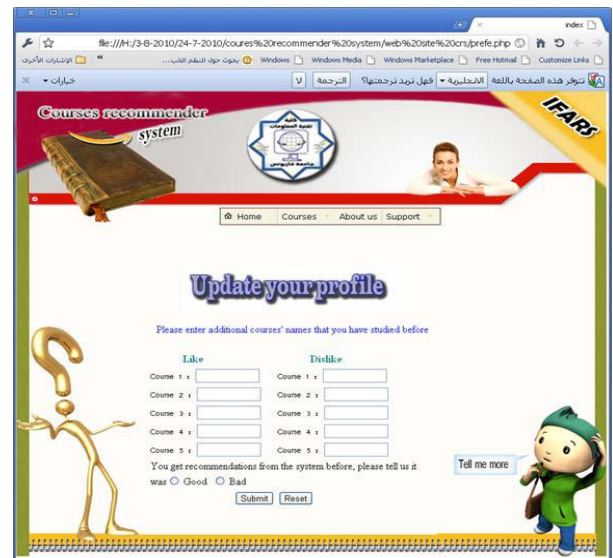


Fig 18. Gather preferences to update the student profile.

5. CONCLUSION

In this paper an intelligent electronic academic advisor has been designed and developed. as tool for academic advising Implementing MAS and recommendation algorithms will increase the efficiency of the learning process in a college system, and a case study was developed and presented in order to demonstrate the scope, functionality and practical application of the academic advising for course registration.

The colleges adding an Intelligent Electronic academic advisor system to their academic advisor system will benefit from the following:

- An Intelligent Electronic Academic Advisor would reduce the workload of advisors and need few advisors within the faculty.
- reduces advising computation time by retrieval items from the different sources at offline stage.
- Reduce the time and effort in involved in academic advising, and students will graduate on time.

There are further advantages in using multi-agent technology to design the electronic academic advising, which are summarized below:

- The scalability of the proposed tool has been examined against the number of students in the faculty. Using agents enriches the proposed tool, making it dynamic, as agent technology has this ability. These agents can easily connect and disconnect from the system.
- The second criterion is the flexibility, which is another necessary factor in dynamic systems. In the context of the proposed tool, flexibility is the ease with which the system can respond to changes without affecting the value delivery.

6. REFERENCES

- [1] NACADA.(2006) concept of academic advising. Retrieved from <http://www.nacada.ksu.edu/Clearinghouse/Advising/Issues/Concept-Advising.htm>.
- [2] Daramola, O., Emebo, O., Afolabi, I.T. and Ayo, C.K., 2014. Implementation of an intelligent course advisory expert system. *IJARAI) International Journal of Advanced Research in Artificial Intelligence*, 5(4).
- [3] Nwelih,E. and Chiemekwe,S., 2012 “Framework for a web-based spatial decision supportsystem for academic advising,” *African Journal of Computing and ICT*, vol. 5, no. 4, pp. 121–126,.
- [4] Martin, H. ,2006"the social cognitive actor : a multi-actor simulation of organisations," PhD thesis, University of Groningen.
- [5] Gandon, F., 2002. "Distributed Artificial Intelligence And Knowledge Management:Ontologies And Multi-Agent Systems For A Corporate Semantic Web," PhD thesis , Doctoral School of Sciences and Technologies of, INRIA and University of Nice, Sophia Antipolis,.
- [6] Stone, P. and Veloso, M. ,1997 "Multiagent Systems: A Survey from a Machine Learning Perspective," *Autonomous Robots*, vol. 8, pp. 345-383.
- [7] Guo, L. , Roberston, D. and Chen-Burger, Y. , 2004 "Business Process Model Based Multi-agent System Development," in *The Second International Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments*, Beijing, China.
- [8] Albashir, A.,2002 Review of Multi-Agent Systems and their tools, university of liverpool, [Online]. Available: www.csc.liv.ac.uk/~ali/wp/MAS_review.pdf [Accessed: 15 SEP. 2018].
- [9] Itmazi, J. ,2005. Flexible Learning Management System to Support Learning in The Traditional and Open Universities, PhD thesis, University of Granada.
- [10] Niinivaara, O.,2004. Agent-Based Recommender Systems, Software Agent Technology Course Paper, University of Helsinki.
- [11] Montaner, M. , López, B. and De La, J. 2003 “A Taxonomy of Recommender Agents on the Internet”, *Artificial Intelligence Review*, vol. 19, Pages. 285- 330,
- [12] Pazzani, M. , Muramatsu, J. and Billsus, D. 1996 “Syskill & Webert: Identifying Interesting Web Sites”, In proceedings of the thirteenth national conference on artificial intelligence, Portland, United States, vol. 1, pages 54-61.
- [13] Resnick P. et al., 1994 “GroupLens: an Open Architecture for Collaborative Filtering of Netnews”, *ACM 1994 conference on computer supported cooperative work*, New York, NY, USA pages. 175–186.
- [14] Tran, T. and Cohen, R. ,2000. Hybrid Recommender Systems for Electronic Commerce, In *knowledge-based electronic markets*, papers from AAAI workshop, Technical Report WS-00-04.
- [15] Burke, R.,2000 knowledge-based recommender systems, In *encyclopedia of library and information systems*, vol. 69.
- [16] Henderson, L. K. , Goodridge, W. ,2015. AdviseMe: an intelligent web-based application for academic advising, *International Journal of Advanced Computer Science and Applications*, 6 (8), 233–243.
- [17] Lin, F., Leung, S., Wen, D., Zhang, F. and Kinshuk, M.2008 “e-advisor: A multi-agent system for academic advising,” *International Transactions on Systems Science and Applications*, vol. 4, no. 2, pp. 89–98.
- [18] Marivate, V., Ssali, G. and Marwala, T.,2008 "An Intelligent Multi-Agent Recommender System for Human Capacity Building," in *Electrotechnical Conference, 2008. MELECON. The 14th IEEE Mediterranean*, 2008, pp. 909-915.
- [19] Nienaber, R. , 2008 ."A Model For Enhancing Software Project Management Using Software Agent Technology," PhD thesis, University Of South Africa