Adaptive multi-agent smart academic advising framework

Abdelaziz A. Abdelhamid and Sultan R. Alotaibi

1 College of Computing & Information Technology, Shaqra University, Saudi Arabia
2 Faculty of Computer & Information Sciences, Ain Shams University, Egypt

Abstract
Academic advising is a crucial process in higher education and usually requires better understanding of student capabilities and curriculum structure to achieve its intended goals. Here, the authors propose a framework of integrated environment based on multi-agents to automate the full process of academic advising. The proposed framework consists of six agents namely, student agent, instructor agent, administrator agent, performance agent, schedule agent, and smart advisor agent. These agents are interacting together with the help of smart advisor agent, which manages the communication between them and provides smart advice based on machine learning techniques. In addition, the analysis of the proposed framework along with the deployment map is discussed by the authors. Moreover, a case study is presented in terms of a sample part of adaptive multi-agent smart academic advising framework to demonstrate the workflow of the proposed approach.

1 INTRODUCTION

The academic success of students in higher education is highly affected by the process of academic advising. However, when the number of students’ enrolments increased, this process becomes time consuming and relatively hard to maintain. On the other hand, this process tends to be ignored by many students, unfortunately, due to the lack of awareness about the importance of this process and the decreased number of advisors/students ratio in many academic institutions [1].

Throughout the tertiary education environment, academic advising is a collaborative mechanism between advisors and students intended to improve the overall educational experience of a student by providing support for academically related concerns. It is accomplished by evaluating the academic records and external influences of the student (academic skills, regular schedules, academic interests, and financial constraints) to create personalized guidance that best matches his/her needs. This guidance will then encourage students to make educational decisions so that they can plan for their academic study to match their life goals by completing their academic degrees within the specified timeline or less [2, 3].

There are several features of the academic advising process such as, long-term, iterative, and time-limited features. Due to these features, academic advisors sometimes cannot provide the optimal advice to students because these advisors cannot predict which courses students may pass or fail to provide them the perfect advice about future semesters. Therefore, this process should be made available at least once per semester to follow up students’ progress and hence ensure that they follow the rules of their academic programs [4–6].

There are four models for academic advising namely, developmental, engagement, prescriptive, and integrated. In developmental advising, a high level of student-independence is fostered through sharing the decision making process between student and his advisor. This process is based on suggesting a set of proper resources to students to help them in making the right decision. Engagement advising is a type of developmental advising in which more student-advisor meetings are involved. On the other hand, prescriptive advising is advisor-centric, in which the decision is solely made by the advisor. The fusion of the former types is called integrated advising [7]. In the colleges of Shaqra University, the prescriptive type is commonly followed, in which the human advisor is the main actor of the advising process.

There are many challenges affecting the practical implementation of the academic advising process in a professional way. From these challenges, considering the academic advising process an extra task added to the other tasks of the advisor. Sometimes, this makes a barrier to the advisor to properly
follow-up the responsibilities of the academic advising process. In addition, when there are many students assigned to an advisor for academic advising, the length of time available for holding meetings with all of them is also affected. Consequently, the advisor could not allocate adequate time for each student to guide and offer him solutions to the raising issues during the study, which negatively affects students’ academic progress.

When academic advising presented to students is poor or in case of no advising, sever repercussions are highly expected to affect their progress in the academic study, and possibly cause a graduation delay. Therefore, academic advisors should have a full knowledge of their students in order to give them an effective advice. To realise that, personal meetings should be held between the advisor and his student in order to strengthen the relationship between them. Consequently, the advisor can understand the unique needs of the student, and thus guide him properly. However, due to the limited availability of committed and experienced advisors, this task becomes difficult to achieve [8, 9].

As the main task of the academic advising process is the selection of courses, two core methodologies can be used to realise this task namely, top-down and bottom-up approaches [10]. In top-down approach, student starts with determining the advanced courses as his initial interest. These interests are analysed, and then the prerequisite courses are derived and presented to student to take in order until reaching the final goal. The bottom-up approach on the other hand, starts with presenting the courses with no prerequisites as initial interest. In next semesters, course selection is based on the previously completed prerequisites [11, 12].

1.1 | Academic advising process in Shaqra University

The prescriptive bottom-up approach is followed in the academic advising process at Shaqra University. In college of computing and information technology, this process is handled by the committee of academic advising under the supervision of the deanship of admission and registration. This committee assigns about 25 students to each staff member for academic advising. Most of these advisors are not highly qualified to be ready for offering professional advice to students. There should be regular meetings between the advisor and his students each semester to handle their academic matters. However, due to the limited availability of the advisors, this process is handled less frequently.

Figure 1 shows the advisory process at Shaqra University. As shown in the figure, the process starts with getting the advising form from the administrative staff then filling it with the requested advice. The academic advisor, checks the form whether it is related to the enrolment or attendance matters, as these are the only concerns active in the academic advising process. If the request is related to course enrolments, in this case, the advisor validates the form data with the student record, and then suggests a proper advice based on the study plan and the university rules. However, in case of attendance matters, the advisor asks the student to provide him with an excuse document (i.e. medical report) along with filling a form dedicated for this matter to get an approval from the department head and vice-dean for student affairs. The advisor then notifies the course coordinator about the attendance excuse of the student. Finally, the advisor returns a proper feedback to the student about his request.

1.2 | Difficulties of the traditional academic advising process

The main difficulty facing the traditional academic advising process is the quality of the academic advising. To improve the quality of this process, dedicated staff are required to receive all students’ requests all the time and handle them through continuous availability. However, this is impractical due to the un-feasibility to hire staff for the academic advising only. Therefore, this task is unfortunately, assigned to teaching staff members as an extra task, which greatly affects the quality of this process due to their intensive labour duties [13, 14].

In addition, the length of meetings held between advisor and his students also affects the quality of the advising process. The large number of students assigned to each staff member for advising makes students get inadequate time with the advisor for their meeting. This case usually happens at the beginning of each semester when the course registration is opened to students [7].
Moreover, the advising process may go through unnecessary hiccups due to the improper representation of the information presented by students, such as the hand filled forms provided by the advisory representatives. Due to the human error expected from students, these forms usually suffer from incorrect information, which usurp useful time from the advisor to validate these information against the student information system. Recently, Shaqra University replaced the paper-based forms with online forms. However, these forms suffer from the same problem as it still depends on manual data filling, which also may contain improper student information. In all cases, the advisor needs to full student record to provide him a proper advice or decision, which is done rarely and manually.

The administrative matters also affect the quality of academic advising. From this perspective, the high quality of advising can be achieved by providing advisors with full knowledge about the degree requirements and study plans. In the real case, study plans can be under review; therefore advisors should be aware of the up-to-date changes applied to these study plans. The lack of information from higher administration may form a barrier to provide high quality advising and cause frustration and dissatisfaction to students, in addition to the potential delay of graduation.

1.3 | AMASIA: The multi-agent framework

The current progress in computer science in general and in artificial intelligence in specific greatly helps providing smart solutions to the academic advising process. This would allow a remote alternative to students who could not hold a physical meeting with their advisor. Practically, these smart solutions will not completely replace the human advisors altogether, but will reduce the advisor’s workload and alleviate his cognitive stress. The smart solution for this process is usually based on handling all student data along with the university regulations when making an automatic decision. This allows human advisor focuses closely on the qualitative matters of students, which improves the quality of the overall academic advising process [2].

In this paper, adaptive multi-agent smart academic advising (AMASIA) solution is proposed. This solution provides an intelligent web-based automatic advisor based on multi-agents with its core built on top of machine learning algorithms. This solution is motivated by the passion of College of Computing and Information Technology, Shaqra University to raise the quality of the academic advising process by employing recent advances in artificial intelligence and machine learning. The proposed solution is planned to automate the full process of academic advising from all perspectives based on multi-agents. However, most of the currently existing systems usually handle the academic advising process from limited perspectives such as; course deduction, timetable adjustment, online submission to advising forms, or online booking of appointment with human advisor [15].

The core of the proposed system is based on utilising the full information needed in academic advising, such as student academic records, university regulations, previous automatic advice, semester schedules, feedback from course instructors and students, and academic programme information, into an adaptive machine-learning paradigm to deduce reliable advice to students. The proposed solution continuously keeps tracking the progress of students and can predict the eligibility of their qualification. When the system provides an advice to a student, but the generated advice is not sufficient and a special human attention is required, the system automatically books an appointment with the human advisor and establishes a remote interaction between them via recorded chatting. The outcome of this direct interaction between student and his advisor is then stored to be included in the analysis of the upcoming student advising requests.

In addition, the proposed framework provides a support to human advisors by allowing them to view all the previously deduced advice, and enables them to place comments on the profiles of students to be considered in the future analysis. This results in a clear history of alright student information. In addition, it supports human advisors with an access to alright forms, plans, reference manuals and documents to be updated when necessary through the corresponding agents.

Moreover, the proposed system is adaptive in which it updates its behaviour automatically based on the feedback and progress status of students. In particular, performance monitoring agent keeps monitoring the student’s progress by tracking the recorded marks, attendance, and percentage of participation, which are provided by course instructor. Therefore, when the progress of student does not meet the expectations, Performance monitoring agent communicates with the smart advisor agent to deduce a proper advice and send it automatically to the student and his human advisor to take an early action to avoid the potential delay in student graduation.

As the proposed framework is composed of complex and distributed environment, it was necessary to maintain all the services provided by the system based on intelligent agents. Each agent is autonomous and can handle a separate service in the system. The collaboration and information exchange between these agents is performed using Knowledge Query and Manipulation Language (KQML) protocol. There are six agents defined in the proposed system namely, student agent, instructor agent, administrator agent, schedule agent, performance monitoring agent, and smart advisor agent. The later agent is the main actor in the system, which controls the core process of academic advising.

This paper is organised as follows. Section 2 presents the literature survey and background of academic advising systems in terms of the strength and weakness of each system. In Section 3, the framework of the proposed AMASIA system is discussed. Section 4 demonstrates the deployment plan of AMASIA framework, followed by a case study in Section 5 to verify the workflow of the proposed approach. Finally, the conclusion and future perspectives are presented in Section 6.
In the literature, there are many computerised solutions implemented by institutions to improve their academic advising process. Each solution is tailored to fit the institutional needs, and usually hard coded to avoid purchasing multiple licences for expensive expert system shells [7]. Automated systems provide a simple transition from paper-based advising to computer-based communication between human advisors and their students through representing data in a computerised form. These systems are still relying mainly on the human advisor to analyse students' information and manually deduce the proper advice to the raising matter.

In the literature, academic advising systems are categorised into single-agent systems and multi-agent systems. Single-agent systems usually support few functions and services such as course suggestion and providing information to students, whereas multi-agent systems provide more services and functionalities including all the functions supported by single-agents. Table 1 presents the recent multi-agent systems and single-agent systems along with the methodology, aim, strength, and weakness of each system.

Based on the preceding survey, it can be noted that the academic advising process consists of several interacting stakeholders with various tasks. Therefore, in this paper, we adopted multi-agents architecture and utilised its strength to realise an efficient and adaptive system in a new perspective for dealing with the complexity, interactivity, and distribution of the academic advising process. Multi-agent technology has been used in a wide range of solutions for advanced learning environments and human-centric computing systems [38]. For the academic advising process, we adopted this technology due to the following reasons:

- The continuous change in the environment of academic advising makes it iterative and should be adaptive.
- As the academic advising process is composed of several modules, such as schedule planning, course advising, and progress tracking, it becomes too complex to handle the tasks of these models in a centralised system.
- The academic advising process is usually based on an architecture, which should be scalable and composed of heterogeneous software.
- The distributed nature of academic advising makes it necessary to utilise the advances in distributed systems and distributed artificial intelligence as a natural and intuitive solution.
- Currently, almost all learning environments become online due the current worldwide spread of the pandemic disease. Therefore, the application of web-based technologies along with multi-agents forms better solution to modelling and developing learning environments.
- The need for distributed learning environment to be dynamic and smart and capable of providing intelligent and autonomous functions such as learning and reasoning to offer the best advice to all academic related matters.

The proposed architecture

The overall architecture of proposed AMASIA system is depicted in Figure 2. As shown in the figure, the system is based on six agents namely; student, instructor, schedule, performance, administrator, and smart advisor, in addition to eight resources and data repositories namely, study plan, semester schedule, instructor profile and workload, student profile, student grades and attendance, course suggestions, and students’ complaints. The smart agent represents the brain of the system that applies machine learning techniques in making decisions and providing advice. These agents are interacting together through a single unified web application. The human actors interacting with the system agents are student, administrator (i.e. department head and human advisors), and instructor, and the interaction is performed through web portals dedicated to each actor.

The features provided by the proposed system are described in terms of the use-case diagrams to describe the underlying functions offered by each agent. The use-case model describes the unit interaction between the external entities and the functions provided by the agent. These external entities are referred to as actors; which may include human users, other agents, other systems, or external hardware.

3.2.1 Instructor agent

The main tasks of the instructor agent are to moderate the course topics and follow up the progress of these topics and
| Approach            | Methodology       | Aim                                           | Strength                                                                 | Weakness                                                                 |
|---------------------|-------------------|-----------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Latorre [16]        | Natural language processing | Students dialog system | This is the first approach that employs natural language processing (NLP) in providing academic advising based on a conversational. It provides multiple services to students. It allows students to search based on NLP | This approach does not take student academic records into consideration. Therefore, its performance is limited. In addition, it requires a complete expression as an input to get an answer, otherwise, it gives error. Moreover, the communication with human advisor is ignored |
| Lawrence, et al. [7] | Expert system     | Course suggestion                             | As this approach provides proper course suggestions, it gains high level of acceptance from students' perspectives | To work effectively, this approach requires a manual entry of students' academic records |
| Gutiérrez, et al. [17] | Learning analytic | Follow-up student performance and provide academic planning | The performance of difficult cases can be evaluated using this approach. In addition, the plan of the next semester is provided | This approach does not generate long-term plan and does not send notifications to advisors. In addition, the sample size used in evaluation is small |
| Riah [6]            | Data mining       | Follow-up student performance                 | As students' progress can be monitored by this approach, it can classify students accordingly | The implementation details are not available. In addition, students' goals and interest are not considered |
| Nwankwo [18]        | Chatbot           | Providing course information                 | It can answer queries with complete and incomplete expressions based on intent detection | The main theme of this work is the development of a Chatbot more than academic advising requirements. In addition, it ignores the role of department head and student records |
| Ho et al. [19]      | Chatbot           | Providing course information                 | It can answer queries with complete and incomplete expressions based on intent detection | This approach does not take into consideration students' constraints and options. It also lacks intensive analysis and functionality |
| Plak et al. [20]    | Machine learning  | Follow-up student performance                 | It generates a weekly report about students' performance. In addition, it can predict students at risk and student dropouts and notifies human advisor accordingly | This approach does not provide students with the proper instruction that can improve their performance. In addition, it cannot provide the reasons of low performance of students |
| Gokhan et al. [21]  | Expert system     | Course and scholarships suggestion            | This approach considers academic records of students to suggest a list of courses to register. In addition, it presents a decision support system to suggest a proper scholarship to students based on their academic interests | It does not provide detailed description about the evaluation results. In addition, the scholarship expert system does not operate properly |
| Eckroth et al. [22] | Rule-base method  | Provide course schedules and advice           | It can answer various queries from students, staff, and advisors. In addition, it can generate academic plan for complex scenarios | The process of generating academic plan for complex scenarios is time consuming. In addition, there is no proper user interface provided |

(Continues)
| Approach           | Methodology                           | Aim                      | Strength                                                                 | Weakness                                                                 |
|--------------------|---------------------------------------|--------------------------|--------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Laghari [23]       | Rule-based and case-based reasoning   | Course suggestion        | The accurate course suggestion based on rule-based and case-based reasoning | It takes into consideration students’ record only. Therefore, unable to suggest courses for complex cases. In addition, the generated reports are not well documented |
| Feighali et al. [24] | Jess rule engine                      | Provide advice to students | The implementation of the system in XML allows it to be expandable. It provides advice based on student queries | The system provides advice without considering students interests and their GPA |
| Esraa et al. [1]   | Rule-based system                     | Provide advice to new students | It provides accurate advice to new students                             | It provides advice to new students only. In addition, it is not integrated with running system in the university |
| Wald et al. [25]   | Expert system and natural language processing | Course suggestion        | It helps students to set an active session with their advisors. In addition, it provides course recommendation to students | As this system does not integrate with a database, it is based on manual submission of students’ records to get suggest courses. In addition, the suggestion of courses is based on answering yes/no questions |
| Arun et al. [26]   | Data mining                           | Student performance follow-up | It can accurately predict students at risk. In addition, it can determine the factors that affect the performance of students | This approach is designed for a small database containing students’ information. In addition, the notifications are not sent to students in the proper time |
| Pooja et al. [27]  | Expert system                         | Course suggestion        | It provides postgraduate students with the proper course suggestion based on their research interests | Although the system considers the research interests in suggesting student courses, it does not take into consideration the student's performance |
| Brian [28]         | Neural network                        | Major suggestion         | It takes into consideration the prior knowledge about student preferences to suggest a proper major to them based on a multi-layer neural network | This approach is lacking the evaluation description. In addition, the role of academic advisor is not clear |
| Ghanem et al. [29] | Expert system                         | Major suggestion         | It takes into consideration the prior knowledge about student interests, and future job to suggest a proper major to them based on a multi-layer neural network | This approach is lacking the evaluation description. In addition, it does not take into student’s performance into consideration |
| Al-Ghamdi et al. [30] | Learning analytic                | Student performance follow-up | It can follow-up students’ performance to foresee their likelihood of success or failure | It cannot generate early alert to students with low performance. In addition, the system cannot handle special cases |
| Alpaha [31]        | Case-based reasoning                  | Major suggestion         | It can determine the suggested major based on a comparison between its courses with other majors’ courses | To generate accurate major suggestion, it requires a large set of cases |
| Ismail, et al., [32]| Data mining                           | Course suggestion        | It can properly suggest courses to students based on their interests. In addition, a real | This system can perform only course suggestion, and the modelling of students |
| Approach          | Methodology                  | Aim                     | Strength                                                                                                                                      | Weakness                                                                                     |
|-------------------|------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| Lonn et al. [33]  | Association rule mining      | Course suggestion       | It can generate a list of suggested courses by looking for similar cases. It was applied to a real dataset                                        | The generated results from this system need to be revised and reprocessing by human advisors |
| Lamiaa [34]       | Collaborative filtering      | Course suggestion       | It used a prior knowledge about similar students’ performance to predict the performance of student on interest. As a result, the courses passed by these students are returned as a list of suggested courses | This approach requires manual data entry. In addition, it suffers from the lack of system evaluation description |
| Ganeshan et al. [35] | Decision support system     | Schedule suggestion     | Bases on optimisation techniques, this approach can suggest schedules to students. This is successful for specific cases                      | This approach does not take into consideration the potential conflict of interests. In addition, it does not provide an explanation of decision reason to students |
| Shatnawi et al. [36] | Multi-agent technologies    | Course suggestion       | It uses students’ behaviour and preferences to suggest the suitable courses for them. In addition, this approach is distinguished by the high degree of expandability and flexibility | This approach does not take into consideration the performance of students in the process of course suggestion |
| Sourabh et al. [37] | Rule-based and case-based reasoning | Course suggestion | It uses a prior knowledge about students’ interests in suggesting the proper major for them to achieve accurate results | It does not take into consideration the students’ performance and the requirements of their academic degree |
|                   |                              |                         | dataset was used to evaluate this system when integrates with the main system in the university                                         | is applied based on limited information                                                     |
the percentage of achievement in the course intended learning outcomes (ILOs). Initially, the instructor agent communicates with the schedule agent to provide it with assigned courses and their predefined goals and ILOs. Based on this information, the instructor agent can follow up the course progress. However, for this agent to perform its tasks properly, it requests human instructor to provide it with the topics taught in a weekly manner. Sometimes, instructors may seek adding or updating course topics to match the state-of-the-art progress in the field of that course. In this case, human instructor can ask the instructor agent to make this addition/modification, and then the instructor agent sends this request to the smart advisor, which in turn forwards the request to the administrator agent for approval. Once approved, the smart agent forwards the approval to the instructor agent and asks the schedule agent to add/modify course topics in the study plan and the corresponding ILOs.

In addition, the instructor agent allows human instructor to record students' attendance and marks during the semester. These attendance records along with students' marks are verified by the instructor agent who sends them to the performance agent for analysis before updating the corresponding student profile. The functions provided by the instructor agent are illustrated in Figure 3.

In case of complaints recorded by student regarding the course assigned to an instructor, the instructor agent is notified with this complaint, and thus can view and show it to human instructor to respond accordingly.
The main task of student agent is to identify the preferences and planning requirements for each student. For fresh students, this agent asks them to select their most preferred career tracks as shown in Figure 4. In this figure, student is allowed to select two tracks for his future career. The first one is the most preferred career track and the second one is the best alternative. Based on this selection, the student agent sends a message to the smart advisor agent to prepare the proper study plan for the student. As the fresh students may not be aware of the details of each track, the student agent provides them with more information about each track in terms of the main topics that will be studied along with the potential job offers expected after graduation.

In Shaqra University, College of Computing and Information Technology offers three majors namely, computer science, information technology, and information system. Each major is composed of three tracks, as shown in Table 2. These tracks are used to determine the actual course plan for each student. Based on the progress of student during his study semesters, the first track may be confirmed or may be changed to the second preferred track. Otherwise, student is notified if he could not achieve the required grades that qualify him to the selected tracks, and thus the smart advisor analyses his grades and suggests a proper alternative track for qualification.

On the other hand, student agent has other functionalities that enable human student to view his/her study schedule for each semester. In addition, it displays the automatically suggested courses to student and enables him/her to confirm these suggestions or select other courses. Moreover, it enables student to add a complaint and view its status and also asks human student to fill questionnaires and surveys automatically generated to measure his/her satisfaction during the study. The functions provided by the student agent are presented by the use-case diagram depicted in Figure 5.

### 3.2.3 Administrator agent

The responsibilities of the administrator agent are conducting course curriculum projection and maintaining the curriculum model. It allows human administrator to enter new courses through a web-based interface. In addition, this agent captures necessary permissions from human administrator and forwards them through messages to the other requesting agents. Moreover, the administrator agent has direct communications with both schedule agent, to ask it to manage the resources that it holds, and student agent, to approve or disapprove students requests. In particular, the communication between administrator agent and student agent is held in the form of a negotiation process that targets maximising the enrolments by keeping the workload in the safe limit.

In particular, the administrator agent communicates with the smart advisor agent to confirm the automatic advice and suggestions offered by the smart advisor before sending them to the requesting agent (i.e. student agent). In addition, the
configuration of the core machine learning algorithms used by the smart agent is performed by the administrator agent.

The administrator agent also communicates with the schedule agent to inform it with new constraints imposed by human administrator on the semester course plan. In addition, some workloads of instructors are provided by the administrator agent and sent via message request to the schedule agent which in turn updates the workloads database and thus behaves accordingly as shown in the use-case depicted in Figure 6. Moreover, the administrator agent provides approvals on the automatically generated decisions related to student registrations and courses schedules. This agent applies the university rules to the requests received from other agents and deduces a decision to the human administrator for confirmation. Once confirmed, the decision is sent to the requesting agent to behave accordingly.

3.2.4 Schedule agent

The schedule agent is responsible for creating proper course schedule automatically at the beginning of each semester. It works based on the information gathered from the related resources such as study plan, instructors' workloads, and course schedules of the previous semesters. The schedule agent communicates with the smart advising agent to provide it with the automatically suggested and requests approval from the administrator agent, which applies the university rules.

In addition, when this agent creates the semester plan, it takes into consideration the previous semesters' plans, the workload of instructors, and the instructors' profiles handled by the performance agent. In some cases, the offered course which was taught by an instructor in the previous semester may be assigned to another instructor based on the instructors' profiles and workload. Therefore, efforts can be saved if this agent is fed with full information required to generate the semester plan automatically.
Moreover, this agent accepts a set of constraints from the administrator agent to be considered in distributing lectures and tutorial sessions over the weekly working days. These constraints include, for example, avoiding setting lecture and tutorial of the same course in the same day, and freeing certain time slot in a certain day(s) for holding college events. Figure 7 illustrates the main functions provided by the schedule agent and the collaboration with other agents, such as smart advisor and administrator agents.

3.2.5 | Performance agent

This agent is responsible for creating and updating profiles of both students and instructors. It gathers preferences, feedbacks, and results to build and update these profiles. Performance agent communicates with student agent to get a feedback from student about the course and instructor and rate both of them. If the instructor is new to the college, the performance agent uses the gathered information to build a new profile for him/her. Otherwise, the gathered information is used to add or modify the preferences of the existing profile of that instructor.

Similarly, the performance agent communicates with the instructor agent to provide it with information about student's progress in terms of grades/marks, attendance, and participation during the semester. This communication is performed in a weekly manner to monitor the performance of students and to decide as early as possible if students need proper advice to cope with any potential learning difficulty (with the help of smart advisor agent). These functions are presented in the use-case shown in Figure 8.

3.2.6 | Smart advisor agent

The smart advisor agent is main actor of the proposed system. It contains the machine learning algorithms necessary for deducing proper advice to students. In addition, it coordinates the communication messages among the agents in the system. The tasks supported by this agent are depicted in Figure 9, and can be summarised in the following:

- Suggests courses to students based on their profile and previously determined career tracks.
- Contributes in the generation of the semester study plan as it analyses the data received from performance agent, student agent, and administrator agent to provide the schedule agent with the set of best courses to offer in the semester plan.
Each platform used in the mentioned portal can be displayed with student-preferred tracks. Several platforms involve agent deployment through ADE (Adaptive Decision Engine) which allows students to manage their issues such as career track changes or course withdrawal. The platform also provides proper advice to students regarding their study progress and track achievement. Moreover, it offers proper advice to instructors about course goals achievements and their students’ progress.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.

4 | SYSTEM DEPLOYMENT

Currently, there exist multiple platforms that support the deployment of multi-agent systems. For the proposed AMASIA system we adopted JADE (JAVA Agent Development Platform) platform, which is one of the most widely used platforms with an active community support [39]. JADE runs on two agent platforms running Linux operating system as shown in Figure 10. The first platform carries out the operations of student, instructor, and administrator agents along with the accompanied resources. However, the second platform is responsible for hosting the tasks carried out by schedule agent, performance agent, and smart advisor agent. In additions, PostgreSQL is adopted to provide the data services for the repositories managed by the agents, along with the Tomcat web services. The user agents run primarily on the first JADE agent platform as depicted in Figure 10. Several databases are employed to support the smart agents. Each database is hosted nearby the corresponding agent. These databases are:

- Study plan: It includes full information about courses and prerequisites.

The machine learning techniques that are used to realise the functions of the smart agent include: Reinforcement learning, Neural Networks, K-Means clustering, Association rule mining, and Support vector regression. These techniques are currently under development and experimental results are planned to be presented and discussed in the next articles.
Instructor profile: It contains the feedback from students about the course taught by each instructor, in addition to the percentage of achievement of the taught courses in each semester.

The experimental results of AMASIA system are planned to be completed by the end of academic year 2021, at the College of Computing and Information Technology as the initial phase before applying it to the other colleges of Shaqra University.

5 | CASE STUDY

In this case study, scenario based prototyping is adopted in which we demonstrate how the proposed system will be adapted to the implementation structure. The presented case study will show the process of course suggestion by the smart advisor and the interaction of students in response to that. The course advising system in this case study has a number of characteristics illustrated in the following:

1. Student profile: There is a profile for each student registered in the system. This profile contains full information about students’ previous registrations, preferences, and performance.
2. Course resources: Internal databases that are employed in the system to host alright data about the courses along with their description and availability, study plans, and semester schedules.
3. Domain knowledge: An ontology that can be used to represent the correspondence between student needs and their suggested courses.

The scenario: A new student requests automatic advising for course suggestion. Initially, the system has no knowledge about the student, and thus will create a new profile for him/her. The first step in creating the profile is to determine the preferred career tracks of this new student. This can be done by asking student to provide the student agent with the best two career tracks as shown in Figure 4. Based on these tracks, student agent communicates with the smart advisor agent and sends to it the selected tracks through a message as shown in Figure 11. Then it sends a request message to the smart advisor agent to return the best matching courses, as shown in Figure 12.

The messages transmitted between agents are based on agent communication language. In this paper, the KQML is adopted as the communication language for message exchange between agents of the proposed system. There are several advantages of this language, such as its independence of other transportation mechanisms (i.e. transmission control protocol (TCP)/internet protocol (IP)), content language (i.e. AgentSpeak), and the applied ontology.

When the smart advisor agent receives student preferred career tracks followed by the request, it sends a message to the schedule agent to request the courses of these tracks, as shown in Figure 13.

Once the schedule agent receives the request message, it searches the study plan along with the previous semester schedules to return a list of the available courses based on the running semester and the career tracks, as shown in Figure 14. In addition, it returns the workload of staff members of each course.

The smart advisor agent then filters the provided list of courses to find out the best courses that match student career tracks and depending on the availability of teaching staff based on their workloads retrieved from the schedule agent. After computing the suggested courses, the smart advisor agent sends the best matching courses to student agent to display them to the student agent, as shown in Figure 15.

Finally, the student agent displays the best recommended courses to student as shown in Figure 16. In this figure, the student can approve or disapprove the recommended course for registration. In case of disapproval, he is prompted to select other courses from the list of available courses already returned by the schedule agent.

5.1 | Comparison with existing systems

To verify the significance of the proposed approach, we compared our system with the single-agent and multi-agent systems that target the same task. Table 3 illustrates the various aspects of comparison. As shown in the table, single agent systems have limited functionality and usually do not address all the details of the academic advising process. On the
other hand, the current multi-agent systems partially address the academic advising process. For example, some of these systems address the course suggestion and providing information to students but did not address the role of the course instructors. Therefore, to the best of our knowledge, there is no integrated system that addresses all the facets of the academic advising process including all the actors contributing in it. This gives the proposed approach a high priority when compared with other approaches.

6 | CONCLUSION

In this paper, AMASIA is proposed to tackle the dynamic and complex academic advising process. The general architecture, analysis, and deployment methodology are presented and discussed in this paper. The framework is based on six agents namely, student, instructor, administrator, schedule, performance, and smart advisor agents. In addition, six data repositories are also employed namely, study plan, semester plan, instructor workload, student marks, student profile, and instructor profile. These resources are allocated nearby the agents to support their operation. The communication between the agents in the framework is performed through knowledge query and manipulation language protocol to standardise the transfer of messages between the collaborating agents.

The proposed framework is planned to be applied to the College of computing and information technology, Shaqra University, Saudi Arabia, as an initial phase. We will continue to work on empirical studies of this initial phase, to test the scalability of the proposed system and the usability of the offered functions and services. Further analysis of the data accumulated in planning and knowledge maintenance processes can be done to assist in determining decision-making behaviour and the suitability of the offered advice by the smart agent. This will make the proposed smart system more responsive and adaptive. Therefore, better services can be presented by the system to students, instructors, and administrators.

ORCID
Abdelaziz A. Abdelhamid © https://orcid.org/0000-0001-7080-1979

REFERENCES
1. Afffy, E., Nasr, M.: A proposed model for a web-based academic advising system. Intl. J. Adv. Netw. Appl. 9, 3345–3361 (2017)
2. Hu, Q., Rangwala, H.: Academic performance estimation with attention-based graph convolutional networks. In: Desmarais, Michel C. et al. (eds.) Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada. International Educational Data Mining Society (IEDMS) (2019)
3. Ishak, I.B., Lehar, M.L.B.: A conceptual framework of web-based academic advisory information system. In: IEEE Symposium on Humanities, Science and Engineering Research, pp. 957–961. Kuala Lumpur, Malaysia, ISSN (2012). June 20122378-9816
4. Alfarsi, G., M Omar, K.A., Juma, M.: A rule-based system for advising undergraduate students. J. Theor. Appl. Inf. Technol. 95(01), 2453–2465 (2017)
5. Mohamed Aly, W., Ahmad Eskaf, K., Serry Selim, A.: Fuzzy mobile expert system for academic advising. In: IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–5. IEEE, Windsor (2017)
6. Eculala-Encarnacion, R.F.: Academic advising system using data mining method for decision making support. In: 2018 4th International Conference on Computer and Technology Applications (ICCTA), pp. 29–34. IEEE, Istanbul (2018)
7. Lawrence, K., Goodridge, W.: AdviseMe: an intelligent web-based application for academic advising. Intl. J. Adv. Comput. Sci. Appl. 6(8) (2015)
8. Cline, B.E., Brewster, C.C., Fell, R.D.: A rule-based system for automatically evaluating student concept maps. Expert Syst. Appl. 37(9), 2282–2291 (2010)
9. Laghari, M.S., et al.: A one-semester course planner for EE students. IRJEECE. 1(1), 13 (2015)
10. Beheishi, M., et al.: Student advising system. In: ISBN:978-1-880094-60-0 Honolulu, Hawaii (2006)
11. Daramola, O., et al.: Implementation of an intelligent course advisory expert system. Intl. J. Adv. Res. Artif. Intell. 3(5) (2014)
12. Al-Sarem, M.: Predictive and statistical analyses for academic advisory support. CoRR, abs/1601 (2016)
13. Kowalski, K., Goetz, J., Alam, M.: Intelligent on-line advising with expert system shell. In: ISBN: 978-1-880094-60-0 Honolulu, Hawaii, USA (2006)
14. Gilabert, E.R., et al.: Developing models for online academic advising: functions, tools and organisation of the advising system in a virtual university. IJTEL. 3(2), 124 (2011)
15. Assiri, A., Almalaise, A.S., Bulesee, H.: From traditional to intelligent academic advising: a systematic literature review of e-academic advising. Intl. J. Adv. Comput. Sci. Appl. 11 (2020)
16. Edward, M.L.-N.: An intelligent natural language conversational system for academic advising. Intl. J. Adv. Comput. Sci. Appl. 6, 110–119 (2015)
17. Gutiérrez, F., et al.: LADA: A learning analytics dashboard for academic advising. Comput. Hum. Behav. 107, 1–13 (2020)
18. Wilson, N.: Interactive Advising with Bots: Improving Academic Excellence in Educational Establishments. American Journal of Operations Management and Information Systems (2018). https://doi.org/10.11648/j.aomis.20180301.12
19. Ho, C.C., et al.: Developing a chatbot for college student programme advisement. In: International Symposium on Educational Technology (ISET), pp. 52–56 (2018)
20. Simone, P., et al.: Early Warning Systems for More Effective Student Counseling in Higher Education - Evidence from a Dutch Field Experiment. In: Conference of Society of Research on Educational Effectiveness (SREE), pp. 1–4. Amsterdam (2019)
21. Engin, G., et al.: Rule-based expert systems for supporting university students. Procedia Comput. Sci. 31, 22–31 (2014)
22. Eekroth, J., Anderson, R.: Tarot: a course advising system for the future. J. Comput. Sci. Coll. 34(3), 108–116 (2019)
23. Laghari, M.: E-course planning software system. J. Software. 13, 219–231 (2018)
24. Feghali, M., Zhibi, I., Hallal, S.: A web-based decision support tool for academic advising. J. Educ. Technol. Soc. 14, 82–94 (2011)
25. Aly, W., Eskaf, K., Serry, A.: Fuzzy Mobile Expert System for Academic Advising (2017)
26. Namhbar, A.N., Dutta, A.K.: Expert system for student advising using JESS. In: 2010 International Conference on Educational and Information Technology, vol. 1, pp. 312–315 (2010)
27. Lodhi, P., et al.: Sua: an intelligent student assistant. Intl. J. Interactive Multimedia Artif. Intell. 1 (Spain) (2018)
28. McManus, B., An automatic dialog system for student advising. J. Undergraduate Res. Mankato, vol. 10(1), pp. 1–11. Minnesota State University (2010)
29. Ghanem, A.S., Alobaidy, H.: Data mining for intelligent academic advising from noisy dataset. In: International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), pp. 1–5. IEEE, Sakhier, Bahrain (2018)
How to cite this article: Abdelhamid AA, Alotaibi SR. Adaptive multi-agent smart academic advising framework. *IET Soft.* 2021;1–15. [https://doi.org/10.1049/sfw2.12021](https://doi.org/10.1049/sfw2.12021)