American Journal of Operations Management and Information Systems
2018; 3(1): 6-21
http://www.sciencepublishinggroup.com/j/ajomis
doi: 10.11648/j.ajomis.20180301.12

SciencePG
Science Publishing Group

Interactive Advising with Bots: Improving Academic Excellence in Educational Establishments

Wilson Nwankwo

Department of Computer Science & Information Technology, Wellspring University, Benin, Nigeria

Email address:
drswilson@live.com

To cite this article:
Wilson Nwankwo. Interactive Advising with Bots: Improving Academic Excellence in Educational Establishments. *American Journal of Operations Management and Information Systems.* Vol. 3, No. 1, 2018, pp. 6-21. doi: 10.11648/j.ajomis.20180301.12

Received: February 15, 2018; **Accepted**: March 1, 2018; **Published**: March 20, 2018

Abstract: Student advising services are often regarded as the mainstream vehicle for promoting relationships, understanding, and performance in academic institutions especially at the tertiary level. However, it is often fraught with challenges in developing countries in respect of insufficient supporting manpower and attendant high cost of running effective services. In institutions where the services exist, not all students benefit from it as a result of some factors such as: the sub-optimal performance of the advising personnel; negative psychological complex in students arising from unusual egocentrism (especially in those students who are regarded as "low performers" and would prefer not to be openly confronted); handicapped students especially those students with visible handicaps e.g. speech problems, etc. This paper is the first part of a study aimed at creating a balance in the foregoing situations by presenting a design of a faceless automated "*AdvisorBot*" based on the bot framework. The design reflects a virtual support system model which could be adopted to enhance student support and course advising efficiency. Analysis of the existing system in most tertiary institutions in Nigeria reveals that student support services actually exist though not efficient in the sense that there are seldom specialized units/departments dedicated to this function in majority of the Institutions especially the public institutions where student advising is the work of academic staff in the various departments. The design follows a mix of the agent and object-oriented approaches and produces an implementation-ready specification whose full implementation would effectively support students during their studies. The system facilitates the process of advising by providing quick and easy access to valuable information, and giving important feedback on several issues involved in student advisement, which otherwise would take considerable time.

Keywords: Academic Advising, Intelligent Bot, Bot, ChatBot, DSS, Tertiary Education, Nigeria

1. Introduction

Academic advising is an important function of any academic institution. Appropriate advising and counselling of students increases their chances of success. Academic advising is a decision making process through which a student is guided on the courses and programmes to opt for in a given learning period usually a semester/term or session, based on his/her current academic profile and potentials. Advisement plays a critical role in student's academic performance and an advisor needs to have certain relevant information to render functional guidance to students. According to [1] advising involves a "dynamic relationship between a student and adviser. At the center is a shared responsibility for a coherent education plan that incorporates personal, social, academic, and career considerations. Advising focuses on helping students identify life goals, acquire skills and attitudes that promote intellectual growth, and become academically successful." Ideally, academic advising is often arrogated to human trained in the field of counseling and career development. This is usually the reason behind the establishment of separate and functional academic advisory and student support units or department in some colleges and universities. According to [2] academic advising is effective academic when it directs students to explore and define their own purpose, goals, and curricular pathways to achieve learning and developmental outcomes. Nevertheless, for such effectiveness to be realized, it behoves the institution to ensure that its curricular plans are coherent, and provides satisfactory programs of study that enhance students' educational goals, career and life aspirations.

In Nigeria, such ideal situations are often rare as most tertiary institutions only allocate student advising functions randomly to the available academic staff in the department i.e.

advising is decentralized and managed by each academic department whether or not the number of the available staff are proportionate to the students. Competency is often relegated to the background as the reality is that student population is shared among the staff irrespective of their expertise. In developing countries, where the cost of education is relatively high owing to poor funding, dilapidated infrastructure, poor governance and accountability framework, unstable government policies, and ultimately corruption, less is done on addressing career development of students enrolled into various programs by way of academic advising.

Information Communications Technologies and its inherent success in various areas of human endeavor have endeared it to policymakers in education as potential vehicles for addressing various educational infrastructural challenges. One of such technologies is the decision support system (DSS).

Rob *et a*l.[3] have defined a decision support system (DSS) as "an arrangement of computerized tools used to assist managerial decision making within a business". Note that a business in their context reflects any organization where decision making is a key function. They identified four essential components of a decision support system viz: data store or decision support system database containing business data and business model data; data extraction /filtering component for extracting and validating the data taken from operational databases and other external data sources; end-user query tool for creating queries that access the database; and end-user presentation tool for organizing and presenting the analyzed data. Note the complexity in the foregoing definition. Prior to the foregoing definition, Holsapple and Whinston [4] have argued that the purpose of decision support systems is to improve the decision making ability of managers by allowing more or better decisions within the constraints of cognitive, time, and economic limits. Thus, the ultimate goal of a decision support system is to support better intelligence and design needed by the decision maker.

DSSs have been around for decades and have played vital roles in different spheres where information constitutes the backbone for driving the decision-making process. DSSs operate on existing historical data and can simplify the functions of the academic advisor. In large and growing institutions of higher learning, future advising needs are likely to escalate especially as the number of student enrolment increases as well as in situations where human advising or "face to face" involvement becomes difficult. Automated and real-time non-human advisors would become invaluable in such situations.

1.1. Statement of the Problem

Usually, the objective of the advising function is to improve students' understanding of their academic career as they undertake their chosen programs in the University, Polytechnic, or College. Often, the ratio of student to academic course adviser in any institution could grow as high as 200: 1 in extreme cases requiring more human resources and effort. Admittedly, the repetitive nature of the advising process can easily wear out a human advisor, giving rise to human errors or bias, the result of which is students' misperceived academic state of affairs and subsequent poor performance. Moreover, human advisors may not be well-disposed or dedicated in catering effectively for an increasing number of students requiring advising. It is also interesting to note that some students usually have personal reservations or personality issues that affect their attraction to human advisors and consequently may shy away from such interactions. An uncommon scenario is that involving students those with speech deficits, and disabilities which though may want to relate with human advisors but consider themselves handicapped in some way and would rather avoid any presupposed embarrassment arising through closer interactions. It has been noted that advising has not been successful as it should be owing to the following problems: lack of participation by faculty, enormous adviser-to-advisee ratios, advising limited to course selection and registration [1] among others. The foregoing problems may contribute significantly to the decline in performance of many students that are caught in the web of such challenges. It is believed that a faceless universal advisor could eliminate certain problems especially those due to preconceived psychological imbalances in students.

1.2. Aim and Objectives of the Study

The aim of this paper is to evaluate the existing academic advising process(es) in tertiary institutions in Nigeria in a view to improving the existing system through a properly conceived Intelligent Bot that offers real-time decision support irrespective of the location and disposition of the student/user. The specific objectives are:

a. To examine the nature of academic advising functions in the tertiary institutions in Nigeria;
b. To evolve a systematic approach towards the enhancement of academic advisory functions having regard to the various factors that influence the success of academic advising such as: life goals, values, abilities, interests, limitations; career goals; selection and design of academic program of study; selection of courses; scheduling classes; etc.
c. To design a virtual real-time Intelligent Bot that can support the academic advising process or act as a substitute for the human advisor;
d. To incorporate "interactiveness" and history taking/persistence in the advisory system using permanent information store.

2. Literature Review

The essence of academic advising is to assist students succeed in their respective academic pursuits and life goals, and to provide information and requisite guidance they need to attain those goals [5]. This support is usually incorporated into the academic units and functions of every institution though may not be visible in some cases. Academic advisors are the students' consultants during their educational career. Pizzolato [6] posits that academic advisors guide students through vital activities relevant for their success in their academic pursuits. According to him, academic advisors assist students in understanding the educational requirements, scheduling the most suitable modules, introducing them to vital resources, promoting leadership and socialization skills, assisting in career development, and helping them find ways to make their educational experience interesting.

2.1. Academic Advising Support System

As in other sectors of human endeavor where planning and operations processes have found DSS beneficial, it is equally envisaged that the introduction of decision support systems in the domain of academic advising would enhance the advising process in overcoming some of the many problems such as the limited number of advisors in contrast to large number of students, indisposition of advisors, social relationship problems on the part of the student and the advisor, psycho-physiological problems on the part of the student (e.g. speech defects, disabilities, etc), etc. According to [5], academic advising provides a complex and dynamic environment and constitutes a challenging experimental test-bed for investigating decision support system issues. A decision support system can use information about students, including such parameters as academic programme and course histories to make a reasonable recommendation on courses or sequence of courses for the next term or terms [5].

The pioneers of educational decision support systems, Cox and Jesse in 1981[5] in their study applied backward scheduling logic of material requirements for planning class scheduling. Their system could determine the modules that can be offered in each term across a multiyear plan. This system brought computerized solutions to the prerequisite rules, but their study was macro in nature. Dinkel *et al.* [7] developed an expert system for course time and venue (lecture halls) scheduling in a specific term. Wehrs [8] developed an expert system that could assist the advisor in evaluating student records and propose course schedules. While the above systems had the merit of promoting the idea of exploiting DSSs in course planning and scheduling, they have been criticized for not being geared towards the student. Deniz and Erzan [9] and Isa *et al.[*10] are among studies that hinged their models on the belief that the competence of an advisor is sine qua non to the success of the advisory process. Their models applied statistical forecasting in predicting the success of students after undergoing course advising. McDonald and Prosser [11] have applied a constrained programming framework that employs rule-based expert system in implementing a student advisory system. Though their model was commendable in that it made a step towards intelligent advising and planning system nevertheless, its objectives were cut short owing to: the volume of knowledge required, the complexity of the advising task and the diversity of factors that are involved in academic advising thus making the use of sequential rules extremely difficult; the dynamic nature of academic programme requirements would turn the updating and maintenance of such systems into a crippling task; the intrinsic disadvantages of rule-based expert systems, namely the opaque relationship between rules, ineffective search strategy and inability to learn [5].

Sandvig and Burke [12] proposed a case-based reasoning model based on the assumption that similar students would have similar course histories. Their prototype uses the experience and histories of graduate students as a template for recommending potential appropriate courses for the current students.

Pokrajac and Rasamny [13] proposed the 'Virtual Expert System for Advising' geared towards avoiding schedule conflicts.

Olawande *et al* [14] had implemented an intelligent course advisory expert system which supported the course advising process in the department of Computer and Information Sciences of Covenant University. However, this system could take all inputs from the user and display a list of recommended courses to be registered in a given semester. At best their system could be regarded as a recommendation system owing to the fact that it lacked appropriate "interactiveness" expected from an intelligent advisory system.

A critical look at all the systems and models implemented so far reveals that they all share one feature that is, they are limited to course registration and selection. Course registration and selection do not in reality translate to enhancement in performance even though there is a tendency that it does contribute towards such. A comprehensive academic advising model should incorporate to a large extent the following: exploration of life goals, values, abilities, interests, limitations; exploration of vocational/career goals; selection and design of academic major or program of study; selection of courses; scheduling classes [1], and learning outcomes.

2.2. Agent-oriented Systems

Academic Advising System (AAS) could be expressed in terms of agent-based system. Agreeably, there is no exact definition for an agent; nevertheless, there are some that are remarkable and accepted across many domains. One of the most widely accepted definition is given as follows: "An agent is an entity which perceives its environment and is able to act, typically autonomously and pro-actively, in order to solve particular problems, whilst remaining responsive to its environment. Relating the above definition to an academic advising environment, intelligent advising software would qualify as an agent, whereas its environment would include students, human advisors, databases, etc. Figure 1 illustrates this concept.

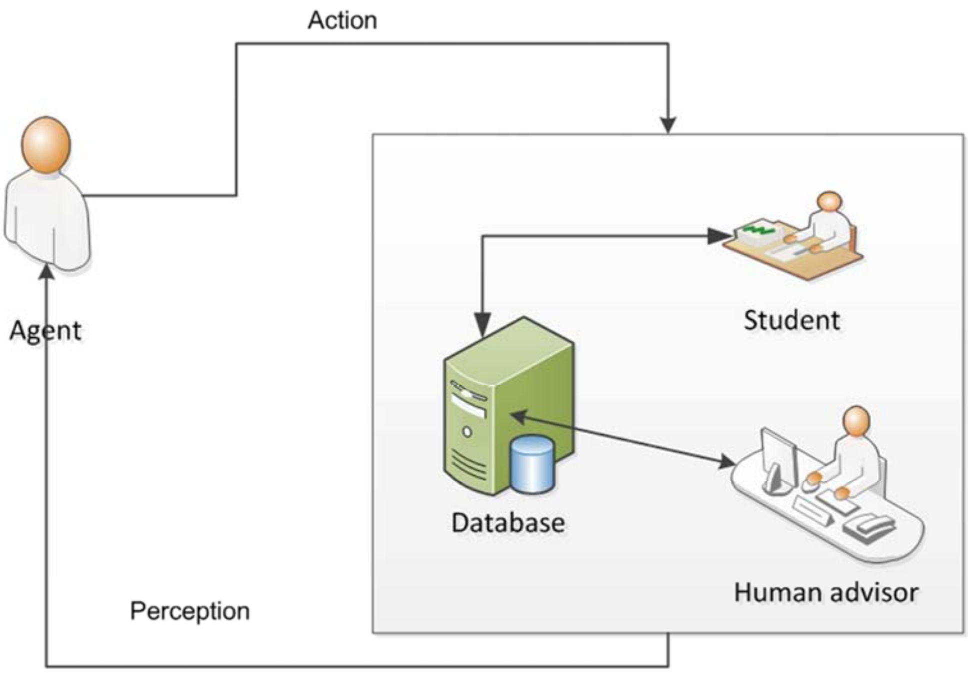

Figure 1. *Agent relationships in an Advising system.*

Agents typically have the ability to interact with other agents, and form cooperating 'societies', or multi-agent systems [15]. Another definition is that given by Wooldridge and Jennings [16], wherein they defined an agent as "a computer system that is situated in some environment, and is capable of autonomous action in this environment in order to meet its design objectives". Jennings [17] defines an agent as an object capable of displaying the following characteristics:

a. Structural Reactivity- Changes in internal structure in response to environmental changes
b. Social Ability: Interaction with other agents through some form of language.
c. Pro-Activity: Goal-directed actions.
d. Autonomy: Some degree of control over its own actions ("self-activation")

Thus an agent differs from an object in that it is capable of self-activation and localized action.

"An agent is a computer program capable of flexible and autonomous action in a dynamic environment, usually an environment containing other agents" [18]. Agent is capable of independent purposeful action. There is a broad consensus that autonomy is a central in identifying agents though this is widely disputed. An agent is saddled with some social roles/responsibilities and interacts with other agents in accordance with the rules associated with the set roles. An agent may be autonomous, proactive, or reactive though it does not follow an agent must strictly belong to one class [19]. An agent is constrained by the roles (also called organizational rules) and behavioural patterns (also called organization structure) thus in this case an agent may be considered as an extension of an object (as in object-oriented systems). Under the agent context, an organization such as a University/College is represented as an entity with a specific structure. The organizational rules may be considered the responsibilities of the organization such as safety rules (time-independent global invariants that the organization must respect) and a liveness rules (that define how the dynamics of the organization should evolve over time) [20]. A role is a specific behaviour to be played by an agent, defined in terms of permissions, responsibilities, activities and interactions with other roles. When playing a role, an agent updates its behaviour in terms of services that can be activated according to some specific pre- and post-conditions.

Usefulness of agents

Agents are widely used in decision-support scenarios where they can perceive/observe current events and make a decision on an event that is consistent with its domain based on available knowledge base as well as take a course of action on the environment. Agents find applications in many areas. These areas are:

a. Virtual/personal assistance [21] [22]
b. Text-learning support [23]
c. Online shopping assistance [24] [25]
d. Game playing [26] [27]
e. Strategic planning support [28]

2.3. Bot-driven Systems

Bot (a compressed word for robot) may be described as a computer program that acts as a virtual assistant in such a way that it simulates a human function in performing a task using natural language. Bots could be intelligent or unintelligent. Intelligent bots are driven by artificial intelligence. Different kinds of bots exist according to the functions they perform. For instance, a ChatBot (which is our focus here) is a kind of bot that provides an interactive platform for people to interact with through a chat interface. The human user can ask questions using voice or text and the ChatBot will usually respond in a conversational style. A

ChatBot may carry out actions in response to the human user's request or response. Other examples of Bots are: Botnets, web-crawler bots, e-commerce bots, etc.

Bots may be classified into different types with respect to functionality, platform, scope of application (e.g. commerce, Productivity, Information, Skill, etc.) etc.

Under functionality, Elharrah [29] has identified seven (7) types of bots. These are:

a. Optimizers (largest class of functional bots-solve concrete problems in a more effective and efficient way than apps and websites;
b. 'One-trick pony' bots- Bots with mini-utilities armed with messaging interface that helps transforms an intention into reality. This class often has impressive cognitive capabilities and extreme viral potentials.
c. Proactive bots: have the ability to provide the right information at the right time and place. Examples include Weathercat, KLM's bot, Poncho etc. all of which are used for very limited applications such as sending solicited and unsolicited notifications.
d. Social bots, which are meant to accomplish a task, exploiting the power of a group or the crowd.
e. Shield**,** a sub-category of Optimizers specifically deployed to help users avoid unpleasant experiences.
f. Chatty Bots: these bots are designed to provide a real-time messaging platform (chat interface) with users for the sake of conversation and engagement.
g. Super Bot: also called intelligent personal assistant due to its ability to make discoveries and resolve issues within a given domain on behalf of a human expert such as an academic advisor

It should be noted that the above classification is not exhaustive hence it does not follow that a specific bot cannot belong to more than one category.

One of the promising classes of bots is the ChatBot. ChatBots simulate a natural person (humans) and can exchange vital information with a human or other bots to the extent of making decisions or recommendations hence could support or substitute a human user in performing advisory functions.

Pratt [30] had identified three types of ChatBots: goal-oriented/transactional bots, conversational bots, hybrid (goal-oriented with strong conversational abilities). A goal-oriented ChatBot helps users achieve tasks e.g. buying a ticket, ordering food, getting specific information, etc. A conversational ChatBot on the other hand, is focused on having a conversation with the user and does not need to deeply understand what the user says and does not have to remember all the context of the conversation. Surmenok [31] had classified ChatBots into business and entertainment bots. According to that classification, business ChatBots are transactional, have a specific purpose, and conversation is usually focused on user's needs e.g. Travel ChatBots provide information about flights, hotels, and tours and helps to find the best package according to user's criteria; Google Assistant provides information requested by the user. In a similar vein, Persiyanov [32] had recognized two major types of chatbots which he called dialogue systems: goal-oriented (Siri, Alexa, Cortana, etc.) and general conversation bots (Microsoft Tay bot).

According to Sansonnet *et al* [33], the functions of a ChatBot may be divided into three: dialogic, rational, and embodied agency functions. As a dialogic agent, a ChatBot must understand the user through natural language interfaces(text, speech, etc.) in order to generate appropriate responses. As a rational agent, a ChatBot must have access to an external knowledge base (database, data warehouse, knowledge store, etc.) and common sense structures to enable it render rational responses to user requests. As an embodied agent, a ChatBot should provide the function of presence so that it could be identified by a given name as in the earliest bots (ELIZA, CHARLIE, etc.).

The block model of a typical ChatBot is shown in figure 2. Functionally, ChatBots work on pattern recognition and a set of algorithms. Structurally, a ChatBot has three main functional components:

a. Entities: The context or domain of the user's request e.g. admission, examination, course selection, product selection, etc. The properties of every entity must be clearly defined.
b. Intents: Questions to be thrown to the ChatBot by the user
c. Responses: The answer(s) provided by the ChatBot.

Intents and entities (Contexts) are the main concepts behind the behavior of a ChatBot. Intents creates links between what a user says and what action should be taken by the bot. Contexts are expressed as string values, and used to differentiate requests that may have different meaning depending on previous requests. A ChatBot would identify appropriate predefined responses from its repository, and then parses it against the appropriate intent and context (entity). The repository is often confined to linguistic building blocks, though its real values may come from structured information extracted from external data sources (databases, etc.).

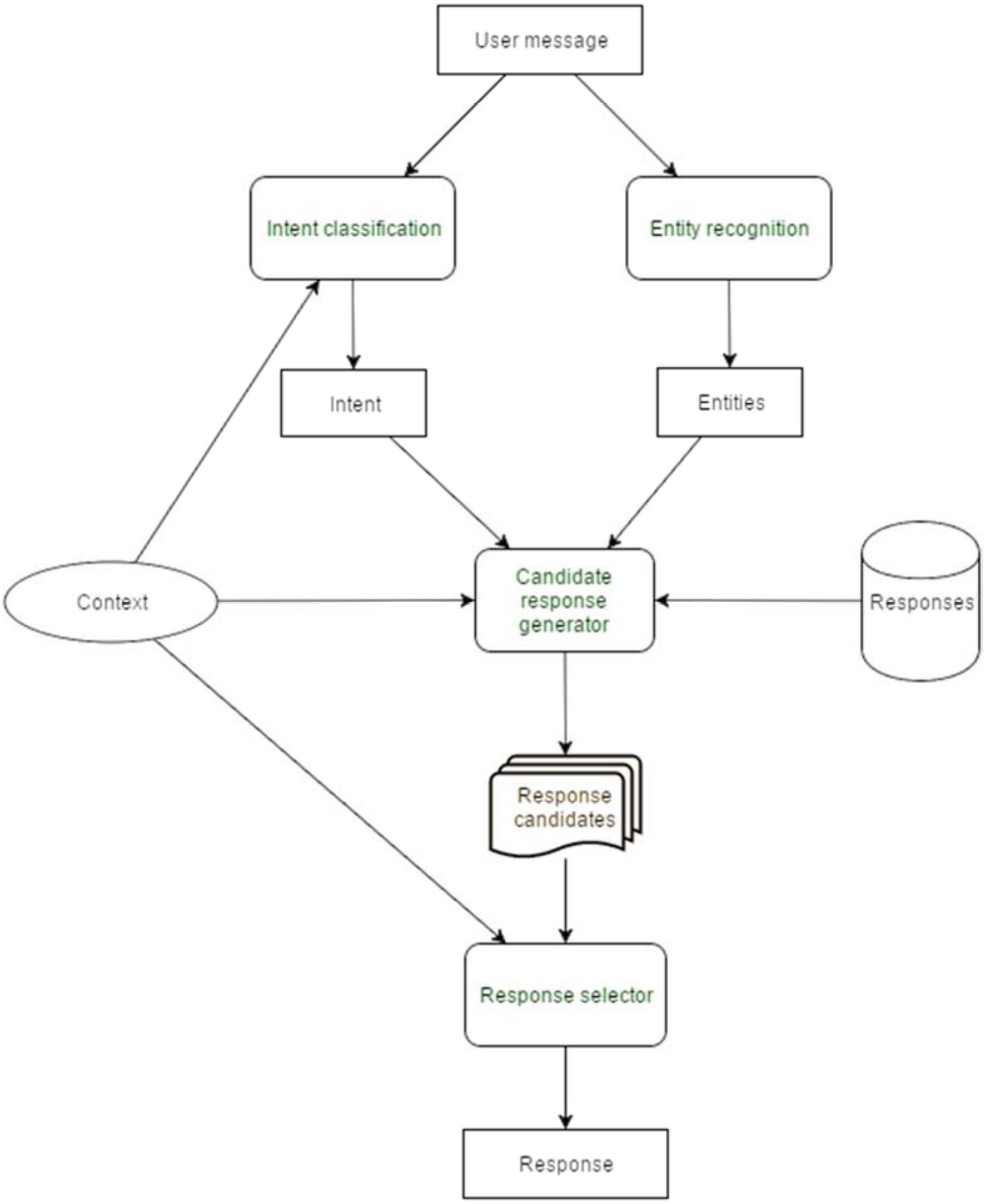

Figure 2. *Block model of a ChatBot.*

2.4. Usefulness of ChatBots

According to Dindayal Gupta of the ChatBot magazine, advantages of using ChatBots [34] are many and some of them are summarized below.

i. Acts as helpdesk with little or no waiting time;
ii. Delivers optimum personalized experience;
iii. Increases customer service level and satisfaction;
iv. Assists in improving user engagement especially when designed suitably, coded and integrated with the business ecosystem, and could be so delightful to users by providing spontaneous service to multiple users;
v. Bots are indispensable especially when targeting a wider audience such as in Universities, Polytechnics, Colleges, medium and large-scale business organizations, government ministries and agencies, etc.
vi. Business intelligence: the built-in artificial intelligence and machine learning capability in intelligent bots empower them in collecting and processing bulk conversational data accumulated over a period as well as deriving valuable, actionable insights from such data.
vii. Improve survey through its real-time discussion and engagement: Bots can be used to conduct survey on experience of customers who purchased or used a product or service regarding whether or not their requirements and expectations were satisfied. Product surveys are often tedious and unrewarding as they are done through emails and websites both of which are known to attract apathy due to various reasons including lack of time.

From the foregoing discourse on bots, it is thus clear that such software components could be designed and employed for effective use in various areas of human endeavor such as student support services where they could assist a human expert or serve as a substitute where the human expert is not available.

3. Methodology

A methodology in the Software Engineering domain may be regarded as interplay between abstractions [35] and tools during the creation of a system. Abstractions in this regards are often based on such elements as data, functions, objects, agents, etc. whereas the tools are used to represent the result of each phase of the software/system development process. Common tools may be divided into formal and informal models. Informal models include: figures; diagrams, tables, text, etc. Formal models include: predicate calculus [36]; temporal logic [37] [38], mathematical objects (sets, functions, tuples, sequences, mappings, axioms), etc. The enhanced object-oriented analysis and design approach was employed in this paper. The enhancement is brought about by the integration of agent-oriented principles into the modeling process. The purpose of this approach is to facilitate analysis, design and implementation, at a level of abstraction that is more adequate for complex information systems, by introducing the single, main concept of agent [39] [40] [41]. The approach employed here exploits the merits of object-oriented approach by expanding the methods and properties by way of adding additional roles, and independence (inherent control of behaviour as against the limited control in objects) derived from the agent-oriented principles.

3.1. Research Design

The data collection followed a cross-sectional study involving a population of 120 persons of which one hundred(100) are students and twenty(20) are academic personnel; and drawn from 20 tertiary institutions that cuts across Universities, Polytechnics, and Colleges of Education, in the Southern and North-central Nigeria. The method of data collection was interview and was aimed at ascertaining the state of affairs on student advising programs in the Institutions. In addition, visits were made to some of the tertiary institutions in the south-south and south-west of Nigeria where the author is domiciled. In all cases, the responses from students and staff reflected the same trend in student advising.

3.2. Materials

Software tools: The software tools include: Visual Studio 2015(for modeling, program design, and implementation), Microsoft SQL Server (backend), AIMLBot2.5, Entity framework, and Microsoft Bot emulator (Microsoft's tool for offline testing of bot application programs). Design was done on a PC with 2.0 GHz quad-core processor with 8 GB RAM and 500GB hard disk.

3.3. Analysis of the Existing System

In Nigeria, the academic curricula in tertiary institutions are built around an academic session comprising two semesters of 13-15 weeks each. However, there is a difference in the duration of the various academic programs. Undergraduate programs run for 3/4/5/6 years depending on the program and mode of entry into the program. The system of course registration depends on each institution, for instance, some institutions allow students to undertake course registration once in a session (that is, all courses for the two semesters that comprise an academic session must be registered at once usually at the beginning of a new academic session), whereas some institutions especially the private ones and few public institutions allow for registration on a semester by semester basis. However, each mode of registration has its pros and cons. Prior to registration of courses at the beginning of a semester or session, students are often directed to consult with their respective academic advisers who are in a position to guide them accordingly. However, academic guidance over the years seems to have been watered down as not much dedication and thoroughness are attached to the advising process.

The traditional approach to student advising common among the tertiary institutions, involves a scenario a face to face meeting by way of a mini discussion is conducted between the student and his/her academic course adviser. Physical contact is important and the major actors are: the student, the course Adviser, and sometimes the head of department. The process of advising may begin with a scheduled or a non-scheduled meeting often initiated at the instance of the student who goes to the adviser to seek advice on issues regarding his/her challenges. The operation of the entire system can be visualized using the activity diagram shown in figure 3 below. From the diagram in figure 3, four distinct phases are identified: verification/validation, inspection, dialog, and recommendation.

3.3.1. Verification and Validation

In this phase, the course adviser verifies that the student is indeed who he claims to be by confirming his identity. He may do this by reviewing the records of all enrolled students in the department, and using the student identification card. Validation may include review of documents relating to settlement of semester fees, departmental dues, etc.

3.3.2. Inspection Stage

This process uses the retrieved student's details (usually contained in a physical file) as input. Such details would include: names, matriculation number, department, etc. The Course Adviser inspects the student records for the previous academic sessions (where such records exist), he reviews the academic profile of the student with respect to performance as to the passed courses, failed courses, dropped courses, and the current academic position of the student.

3.3.3. Dialog Stage

This stage represents a dialog between both actors (student and course advisor. The issues constituting the main stream of dialog may include choice of electives, dropped courses, previous performances, etc. This subsystem continues until an optimal recommendation is reached based on the student's previous academic position, departmental requirements, student's current preferences and the Course Adviser's experience. The output from this subsystem is an adjusted course recommendation.

3.3.4. Recommendation Stage

This is the final phase of the advising process. It uses the output of the recommendation stage as input. Recommendations are documented on a "course registration form" and endorsed by the student and Course Adviser respectively. This endorsement would show that the course adviser approved the selection of courses to be registered by the student. The endorsed course registration form is a byproduct of this phase.

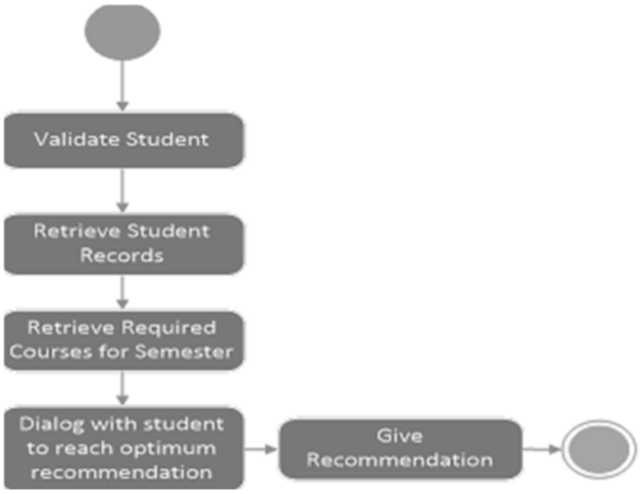
Validate Student
Retrieve Student Records
Retrieve Required Courses for Semester
Dialog with student to reach optimum recommendation
Give Recommendation

Figure 3. *Existing Student Advising Activity Diagram.*

3.4. Problems of the Existing System

It was observed that the existing system is manual and basically restricted to course selection and registration. Whereas this is important, it is believed that modern academic advising goes beyond course registration and selection but includes other important elements such as learning outcomes, abilities, career profiling, etc.

3.5. Overview of the Proposed System

The proposed system is a real-time web-based Bot that employs unsupervised learning through data mining. Every user intending to interact with the system must be recognized by way of a unique identification mechanism such as matriculation number and password (for students) and *staff ID* and password (for staff). The user opens a URL to the system and thereafter supplies appropriate credentials which are verified through iteration of records in the database. On successful authentication the bot displays a welcome message including the previous histories. On successful validation, the bot inspects the student's records including performance profiles from the time of admission, outstanding courses, areas of peak performance, previous recommendations, directives from human advisers, areas that currently require more effort, current learning outcomes, etc. It then uses the information gathered to interact with the student on issues relating to career development, study challenges, class scheduling, assessment of quality of teaching, extra-curricular schedules; etc. At each stage of the interaction, the student is free to supply his request by typing in text values into the input box. Following such input by the student, the system provides a review of student's strengths and weaknesses; provides a direction to be followed which the student may align with. Every request made by the student and corresponding response are recorded in the advising history and persisted against the database. At the end of the man-machine conversation, a log is maintained with a status "awaiting approval" from a human advisor. The second stage involves the human advisor logging onto to the system in same manner as the student. On logon, the advisor sees a list of students whose recommendations await his approval. The advisor could also view the history of the student-system conversation as well as the student's academic profile. The human advisory role is optional in that it is only a certification role i.e. the human advisor verifies what the bot has chosen and subsequently approves or adjusts it.

Figure 4 shows the conceptual framework of the proposed system. In Figure 4, academic advising is conceptualized from three dimensions; the course characteristics, performance characteristics, and personality characteristics. The course characteristics include all the necessary parameters associated with course selection and registration; the performance characteristics are not based on course selection but have to do with somewhat extraneous attributes such as: interests, extracurricular activities engaged in, learning outcomes(passes and failures), class scheduling, etc. The personality characteristics are intervening factors and include: student's current preferences towards courses offered in current semester, goals, abilities, choice of career, etc.

The dependent variable is the performance characteristics while the independent variable is course characteristics. The intervening variable is the personality characteristics. The performance characteristics as applied to a student is arrived at by considering the previous learning and class scheduling outcomes and well as the quality of teaching recorded. The use case diagram in Figure 5 shows how student performance is computed. The quality of teaching is computed from the responses previously supplied by the student on quality of teaching. The personality characteristics consists the personality domain typically reflecting a student's likely predispositions such as: preferences; life goals; abilities; choice of career; academic program satisfaction gauges, etc. The intervening variables are to be captured using a combination of an Artificial Intelligence Markup Language (AIML) store and a database.

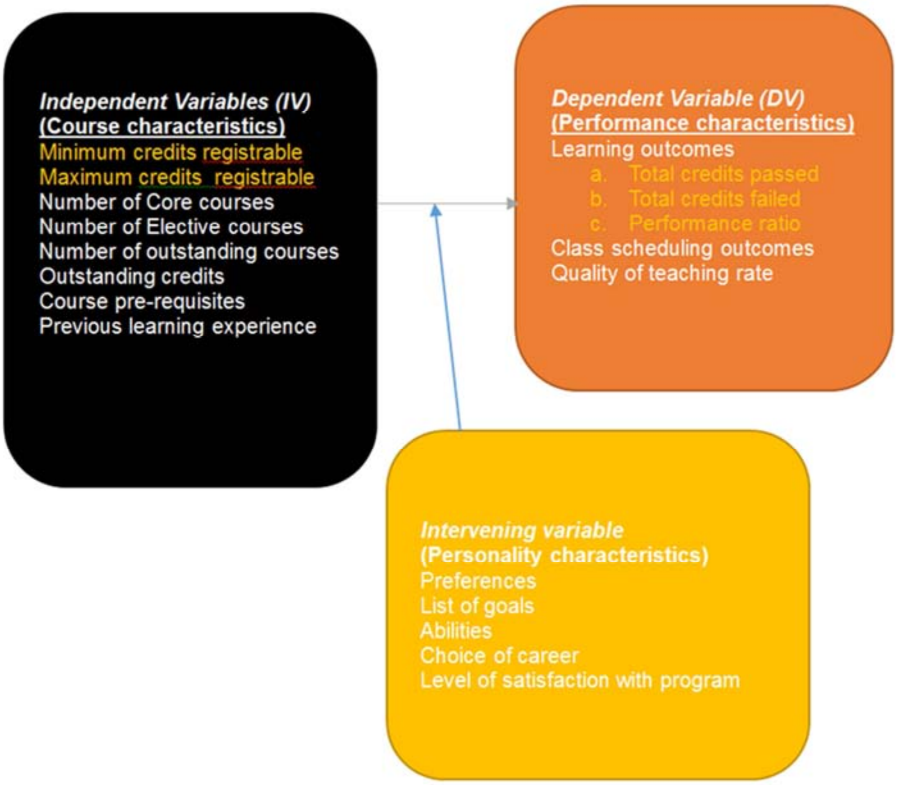

Figure 4. *Conceptual framework of the proposed system.*

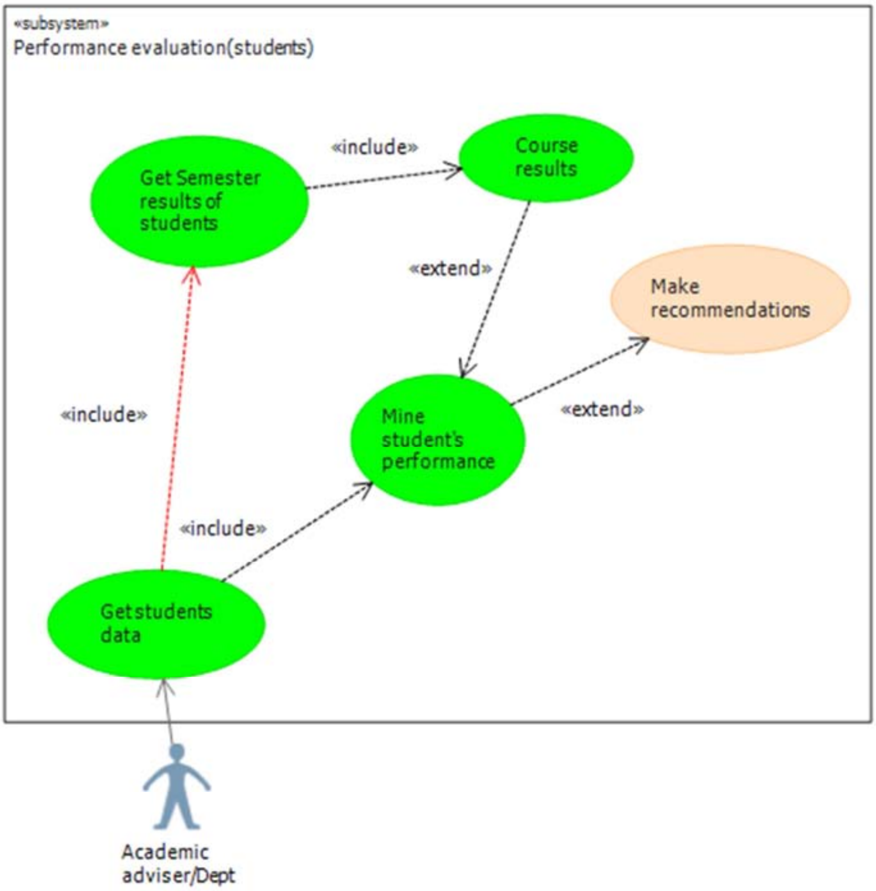

Figure 5. *Student performance evaluation use case diagram.*

3.6. Analysis of the Proposed System

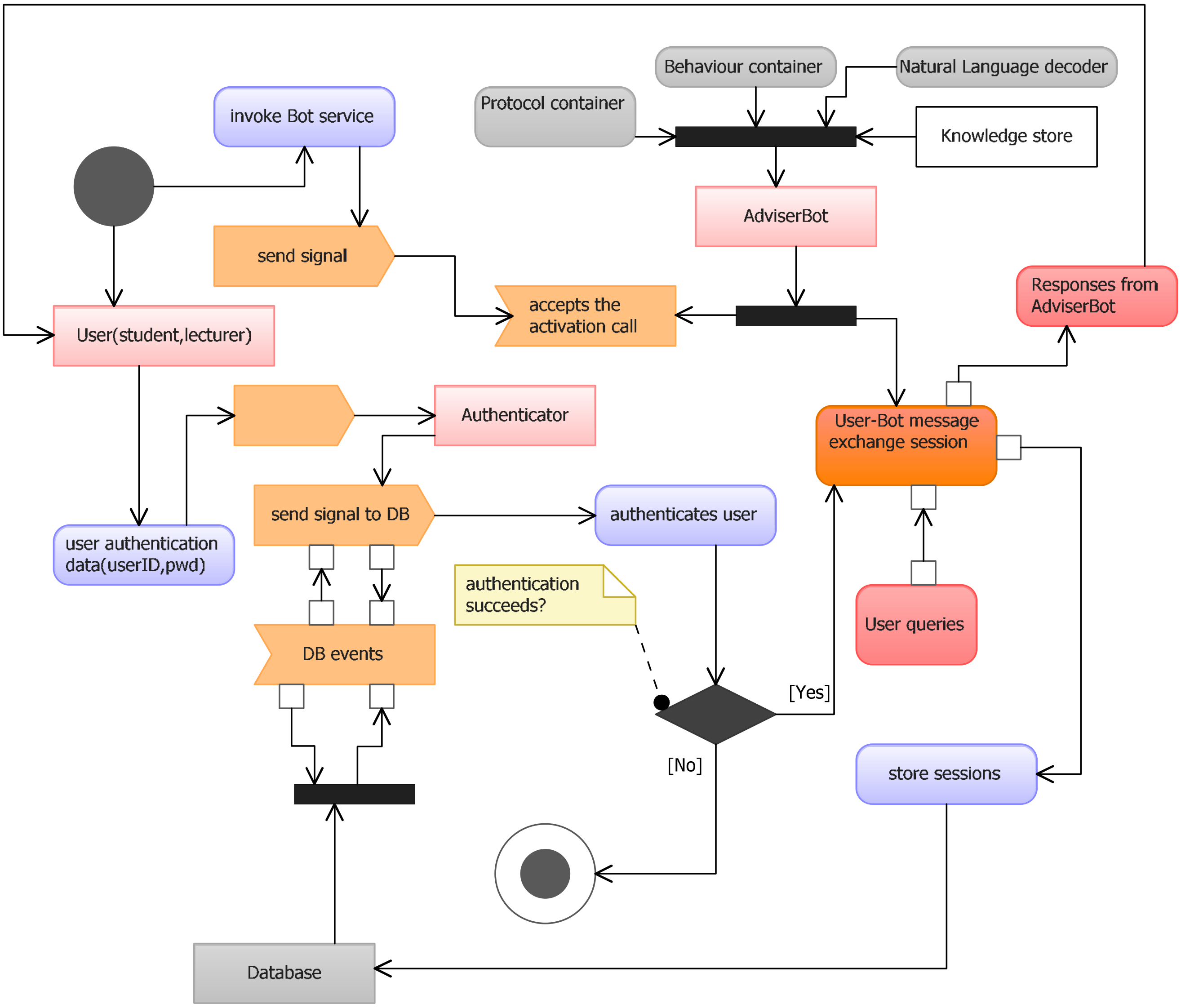

Figure 6. *Activity diagram of the Advising system.*

Three components are vital in the analysis of an agent-based system viz: Agent(s), Behavior (rules) and Metrics at the aggregated level. Object-oriented analysis uses a dozen of tools such as activity diagrams, use case diagrams, class diagrams, component diagrams, etc. to model these components. It does not follow that all object-based tools must be used in the analysis process. Object-based tools employed here include: activity diagrams, and use case diagrams. Figure 6 shows the activity diagram of the proposed system. In Figure 6, there are three main actors: the User; the AdvisorBot; and the Authenticator. Like the AdvisorBot, the Authenticator is an agent-oriented component that is invoked to verify and validate the authenticity of the user prior to engaging the AdvisorBot in conversation. The AdvisorBot is equipped with rules (protocol), deterministic behaviour, knowledge store, and natural language understanding abilities. The knowledge store is built around the following information: previous learning outcomes, performance, preferences; life goals; abilities; choice of career; academic program satisfaction gauges, etc.

4. System Design

The proposed system may be split into different components: the database (backend), AdvisorBot (front agent), Knowledge Base, Cognitive module, etc. The inputs, transformation and outputs revolved around these components. The product of the design phase is a set of models that described the system structure, made necessary trade-offs, provided a blueprint for implementation and allowed the iterative evolvement of the system under controlled conditions.

4.1. Logical Data Design

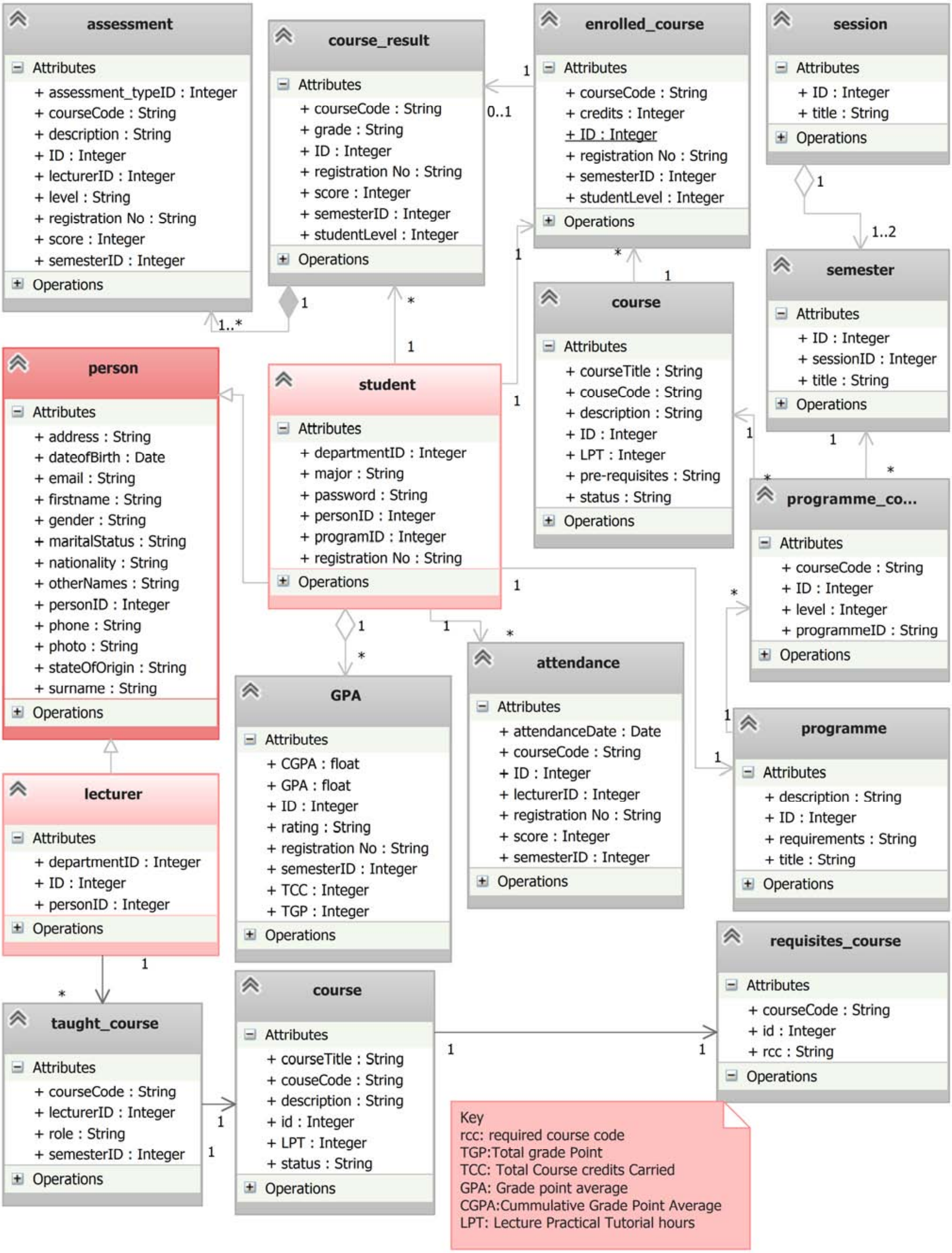

Figure 7. Class diagram representing logical objects and data characteristics.

Two classes of data stores are involved in the proposed system: database data and knowledge base. For the database component, class diagrams were used to represent the various objects that store some relevant data needed to be returned during the conversation process. Data on objects shown in the class diagram in Figure 7 are vital to the determination of course and performance characteristics respectively as stated in Figure 4 above. For instance, data on the minimum and maximum credits to registered by a student in a given semester, the core courses, elective courses, outstanding courses, outstanding and total credits required to graduate, course pre-requisites, etc. are to be extracted from the objects in Figure 7.

4.2. Component Model of the System

The component model is used to show the various operational components of the system. Figure 8 shows the component object model. The model has five main components: the AdvisorBot, query engine, authenticator, knowledge base, and database. Embedded in the AdvisorBot is an inference engine that accepts queries or intents (questions) supplied by the user through the query subsystem. The intents and entities are the basis for the responses given by the AdvisorBot. In order to produce viable responses, the knowledge about the domain of student advising in an academic environment is held in the knowledge base which is linked to a natural language processor component of the AdvisorBot subsystem.

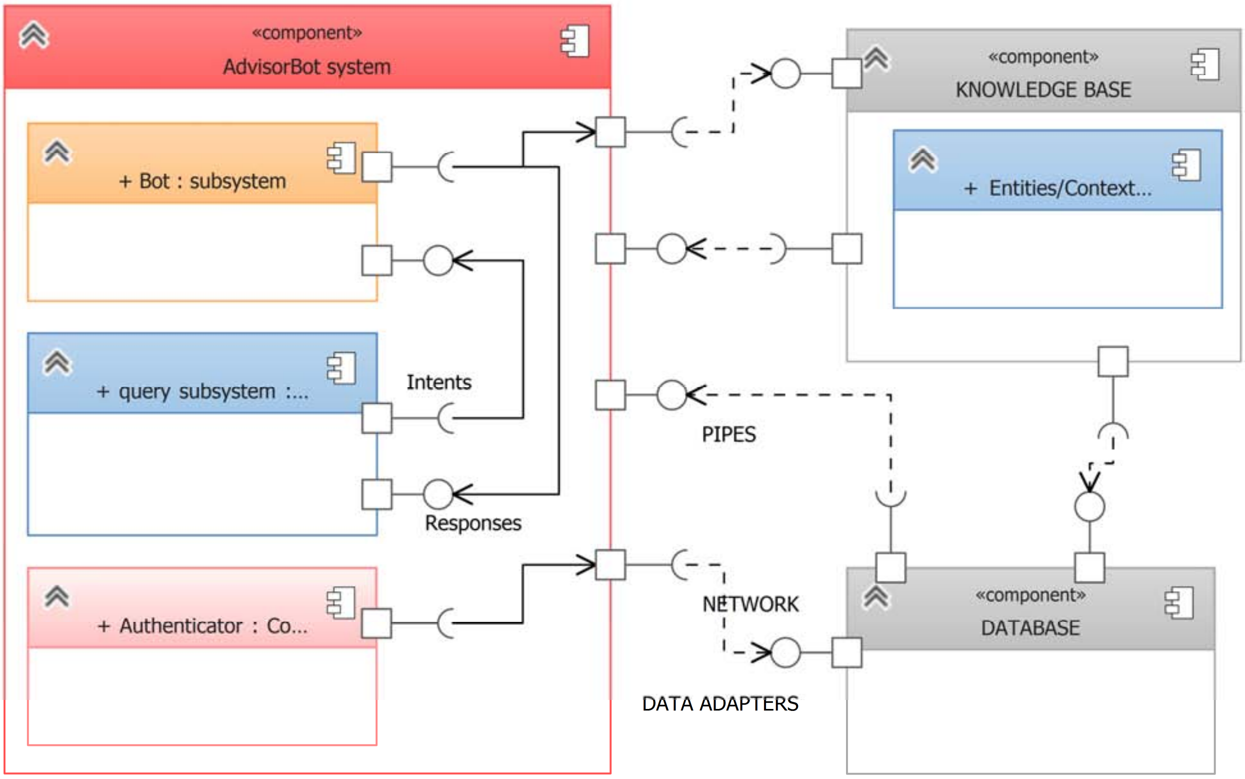

Figure 8. *Component model of the system.*

4.3. Input Design

Input specifies what the user enters into the system to process. The inputs (questions or intents) are built around context/entities and utterances. An utterance is the actual textual input that the user will supply through the chat interface. The entities will capture and pass important information that fulfils the user's request to processing module. For example, in the utterance, "I want to know my CGPA", the entity is the current "CGPA" of the student but defined by "*CGPA_current*", so the language processor would interpret the utterance ("I want to know my CGPA") as "I want to know my *{cgpa_current}*. Then *cgpa_current* which represents the student's current CGPA will be retrieved from the database and presented to the student in the chat interface. An entity such as quality of teaching recorded previously against a course is an important factor in determining whether or not a student is likely to be attracted to learn a new course to be taught by the same lecturer who had earlier taught a course supposedly unsatisfactorily. Such an entity may be associated with the utterance: "How interesting is the *Course* Lecturer?" or "How interesting are the lectures on the *Course* likely to be?" or any of its variants. The domain of the knowledge set encloses all possible properties related to a student's academic career progression (ACP). The ACP is defined by means of Cognitive Services *QnA*(Question and Answer) Maker. The *QnA* provides rich question-based data source from which users can query the AdvisorBot. The questions and answers are enclosed in a file using the Artificial Intelligence markup language (AIML).

Through the *QnA* Maker, a road map is created for the AdvisorBot users so they can ask intelligent questions that can then be re-routed to the database and knowledge base respectively. The input logic is illustrated in the sequence diagram in figure 9 below.

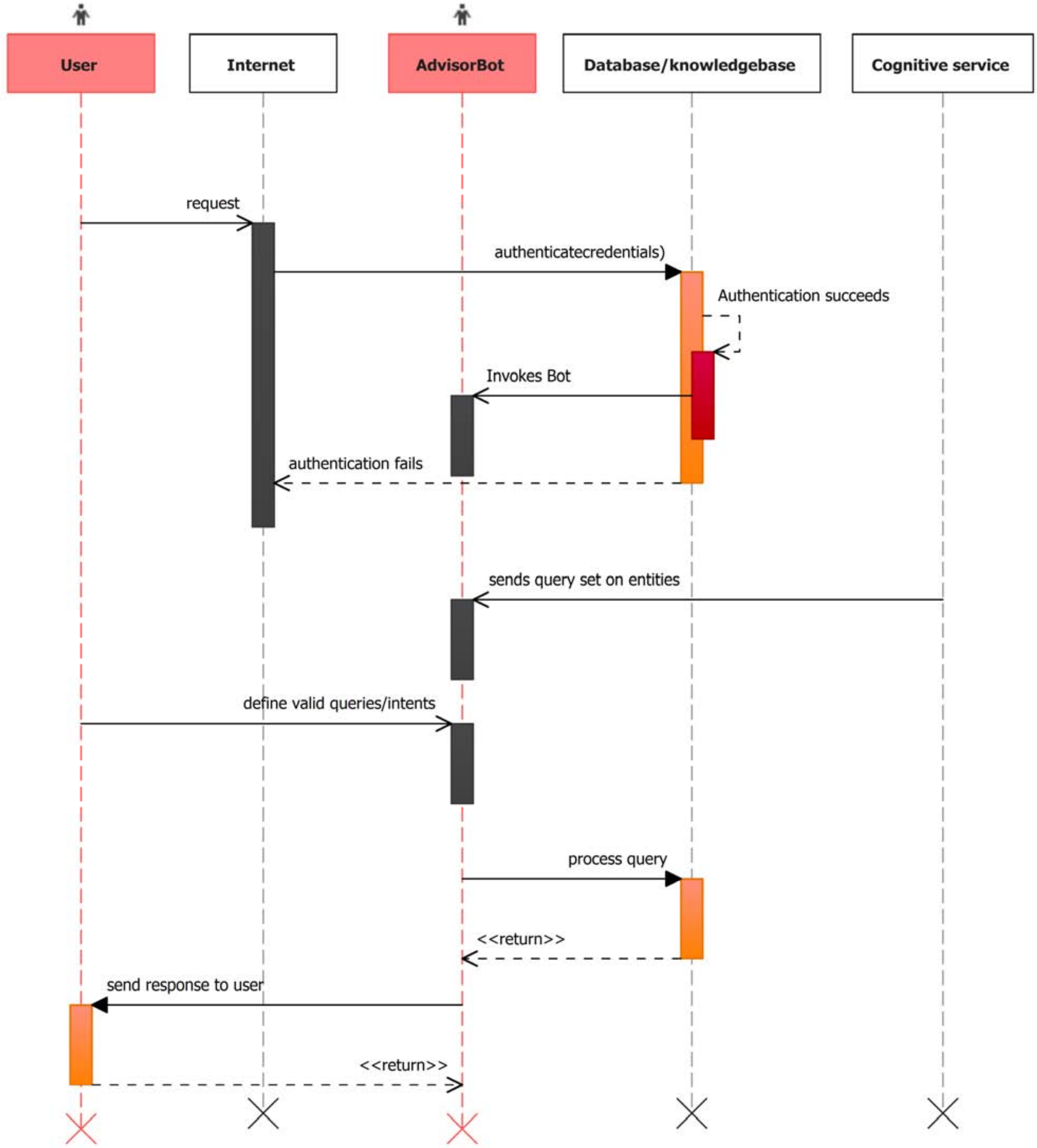

Figure 9. *Input logic.*

Intents design

Table 1 shows the definition of sample intents defined using the AIML and stored in an XML script. Intents are very important because their appropriate identification would provide a sufficient realization of the objective of the *AdviserBot* system. This is because intents generally reflect what the end-users (especially students) would want to accomplish during their interaction with the the system. The intent domain is contained in the table below. The intent tag represents a compressed notation for expressing the functionality to be expected from the *AdviserBot. Accordingly, t*he description of the intent tag briefly describes the various intents that can be channeled to one response. The response in question is the reply the "AdviserBot" would send to the user. Table 1 only shows a crop of intent descriptions.

Table 1. *Intents specification.*

Intent tag	Description
Greetings	Defines all variants of greetings e.g. Hi, hello, Hi there, etc.
Exchange Pleasantries	Defines the various grades of pleasantries. E.g. How are you today?, How are you?, How do you do
Inquire Courses For Semester	What are the *courses* for this *semester*?
Know Course Pre Requisites	May I know the *pre-requisites* for this course? What are the *requirements* for this *course*?
Advise Course Selection	Advise me on what *courses* to *enroll* this *semester*?
Inquire Minimum Credits Registrable	Which *minimum course credits* can I enroll this semester? I need to enroll the *few courses* this semester?
Inquire Pending Courses	What *courses* are required to *graduate*?
Advise Improve Performance	How do I *improve* on my present *performance*? I need advice on *improving* my academic *performance*
Show Outstanding Courses	May I know my *carry-over* courses?
Know Performance History	Tell me about my performance *history*?
Advise Performance	How is my *performance* so far? Have I been *performing* well in the program?
Register Courses	Which *courses* do I register?, Which *courses* am I to enroll?
Know Course Lecturer	Which *lecturer* is likely to take this *course*?
Ask Course Lecturer Ability	What do you say about the *course lecturer's abilities*?
Know Course Interestingness	How *interesting* is this course? Is this course *interesting*?
Know Maximum Course Credits	What are the maximum course credits for this *semester*?
Recommend Schedule In Semester	Suggest a *study schedule* for this semester? Advise me on *study schedule* for this semester
Suggest Extra Study On Course	What are the *requirements* to excel in this course?
Advise Future Performance	Advise me on how to improve my future *performance*
Suggest Electives	Which *electives* do you suggest I enroll this semester?
Precribe Career Goals	Could you define some career *goals* for me?
Advise On Abilities	What do you say about my *abilities* so far?
Suggest Career Line	What do suggest about my future *career* after school?
Ask Program Offerings	What can this *program* offer? What are my future chances on this *program*?
Fare Well	Thanks for your support, Bye! Good bye!; See you next time!

5. Discussion of the Design

Several approaches and platforms are available for designing Intelligent Chat Bots. However, the crux of the matter is the application domain of the intended Bot i.e., scope of coverage. In this paper, emphasis is on evolving a design for implementing an *AdviserBot* that realizes the stated objectives as enveloped by the conceptual framework presented in figure 4 above. The enhanced object-oriented approach was used to specify the various objects that are to be persisted at the backend which in effect provide the necessary data to support the responses given by the *AdviserBot*. These objects including their data characteristics were defined in figure 7 and are to be collapsed into the relevant tables in a physical database. It should be noted that the said objects are also the basis for the entities around which the intents and utterances would revolve. In a similar vein the knowledge base is defined using intents and such intents are captured using the AIML and stored using an Extensible markup language data store (file). The intents displayed in Table 1 are not exhaustive. Intent may have more than one variant all of which should be represented appropriately in the AIML file. The intent may contain keyword(s) which is used for further processing. For instance, when a student sends a message that looks like: "*What do you say about my academic performance?"* the keyword, "performance" would be routed to the logic processing component. This component navigates through the database to access the various records of the student in his enrolled programme. Armed with such data, the bot can return responses that indicate any of good, average, or poor performance rating of the student. For instance, the response could be: "*John, Your performance so far has not been encouraging, having regards to your CGPA of 1.20*". Note that the *AdviserBot* calls the student by name and this is achieved through state maintenance through the chat session. It is ideal for the AdvisorBot to request for the student's registration number earlier in the chat which is stored and used for accessing the student's vital information as long as the chat session lasts. The discussion on the implementation of this design is discussed on the second part of this article.

6. Conclusion

The global awareness on robots (known as bots for short) and the paradigm shift towards the construction of artificially intelligent ChatBots suggest a future of a personalized fluid user experience wherein intelligent systems would perform at a level similar or comparable to humans in interactive messaging. ChatBots compared to other systems are more likely to be more cost-effective channels to deliver content that is contextually aware hence the likelihood of employing them in various domains.

In this paper, the focus was on academic advising bots. Discussion was made on the various developments so far recorded in this field of academic advising and regard was had to the use of plain decision support systems, agents, and bots respectively. An enhanced object-oriented approach supported by a cross-sectional study was applied in formulating the various specifications that are ideal for implementing a versatile advising system. The model was tagged "*AdviserBot*" and hinged on knowledge base and a database. The approach adopted in the design of an intelligent bot is considered straightforward and economical having regard to the fact that no material used required absolute licensing. Though this design is elaborate, it is simply a means to an end. The second part of this paper would detail more on the implementation of this design. It is expected that on implementation the real benefits of using intelligent bots in support academic advising would be realized more efficiently and effectively.

References

[1] Freeman, L. C. (2017) “Establishing Effective Advising Practices to Influence Student Learning and Success”[online]. Available at: https://www.aacu.org/publications-research/periodicals/establishing-effective-advising-practices-influence-student [Accessed 20 January 2017].

[2] NACADA (2017) “The Power to Transform Higher Education Through Excellence in Academic Advising” [online]. Available at: http://www.nacada.ksu.edu/Portals/0/Resources/Excellence%20in%20Academic%20Advising/documents/NineConditionsofExcellence.pdf [Accessed 25 January 2018].

[3] Rob, P., & Coronel, C.(2012). Database systems: Design, implementation, and management (10th ed.). Boston: Thompson.

[4] Holsapple, C. W., and Whinston, A. B.(1996). Decision support systems: A knowledge-based approach. New York: West Publishing company.

[5] Werghi, N. and Kamoun, F. (2010) “A decision-tree-based system for student academic advising and planning in information systems programmes”, International Journal of Business Information Systems, Vol. 5, No. 1, pp. 1–18. Available from: https://www.researchgate.net/publication/220412689_A_decision-tree-based_system_for_student_academic_advising_and_planning_in_information_systems_programmes [Accessed Jan 19 2018].

[6] Pizzolato, J. E. (2006). Achieving college student possible selves: Navigating the space between commitment and achievement of long-term identity goals. Cultural Diversity and Ethnic Minority Psychology, 12(1), 57-69.

[7] Dinkel, J. J., Mote, J., & Venkataramanan, M. A. (1989) “An Efficient Decision Support System For Academic Course Scheduling”, Operations Research, Vol. 37, No. 6, pp. 853-864, INFORMS.

[8] Wehrs, W. E.(1992) "Using an Expert System to Support Academic Advising", Journal of Research on Computing in Education, Vol. 2, No. 4.

[9] Deniz, D. Z., and Ersan, I.(2002). An Academic Decision Support System Based on Academic Performance Evaluation for Student and Program Assessment, The Int. Journal of Eng. Education, Vol. 18, No. 2, pp. 236-244, Expanded SCI journal.

[10] Isa, K., Mohamad, S. and Tukiran, Z. (2007) ‘Development of intelligent planning system (INPLANS): an analysis of student’s performance using fuzzy systems’, Proc. IASTED Int. Multi-Conference: Artificial Intelligence & Applications, Innsbruck, Austria, pp. 567–572.

[11] McDonald, K. and Prosser, P. (2002) ‘A student advisory system: a configuration problem for constraint programming’, ECAI Workshop W4 on Configuration, pp. 20–22.

[12] Sandvig, J.J. and Burke, R. (2005) ‘AACORN: a CBR recommender for academic advising’, Technical Report, TR05-015, DePaul University.

[13] Pokrajac, D. and Rasamny, M. (2006) ‘Interactive virtual expert system for advising’, Frontiers in Education Conference, pp. 18–23.

[14] Olawande, D., Onyeka, E., Ibukun, A., Charles, A.(2014) “Implementation of an Intelligent Course Advisory Expert System”, International Journal of Advanced Research in Artificial Intelligence, Vol. 5, No. 4.

[15] Griffiths, N. & Lim Choi Keung, S. (n.d.) “Agent-Based Systems: An Introduction to Agents and an Overview of Recent/Current Research” [online]. Available at: https://warwick.ac.uk/fac/sci/dcs/research/ias/events/past/ias-griffiths.pdf [Accessed 10 February 2018].

[16] Wooldridge and Jennings, Intelligent Agents. KER 10(2), 1995].

[17] Jennings, N. R.(2000) “On agent-based software engineering“ Artificial Intelligence, Vol. 17, pp. 277-296.

[18] Luck, M., McBurney, P., Shehory, O. and Willmott, S. (2005),”Agent Technology: Computing as Interaction A Roadmap for Agent Based Computing”[online], AgentLink II. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.480.3698&rep=rep1&type=pdf [Accessed 5 February 2018].

[19] Giorgini, P. & Henderson-Sellers(2005) "Agent-Oriented Methodologies:An Introduction"[online]. Available at: http://www.troposproject.eu/files/Henderson01.pdf [Accessed 10 January 2018].

[20] Garoui, M. Mazigh, B. El Ayeb, B. and Koukam, A.(2014) Towards To An Agent-Oriented Modeling And Evaluating Approach For Vehicular Systems Security, International Journal of Information Technology, Modeling and Computing (IJITMC) Vol. 2, No. 1. Available at: https://arxiv.org/ftp/arxiv/papers/1403/1403.2821.pdf [Accessed 21 January 2018].

[21] Kazma, J.(1998) "Intelligent agents: Getting others to do your work for you". IEEE Potentials, Vol. 17, No. 2, pp. 16-19.

[22] Maes, P.(1994) "Agents that Reduce Work and Information Load", Communication of the ACM, Vol. 37, No. 7, pp. 31-41.

[23] Mladenic, D. (1999) "Text-Learning and Related Intelligent Agents: A Survey", Intelligent Systems, Vol. 14, No. 4, pp. 44-54.

[24] [Subramanian, K. R., Shiang, T. K., Lee, S., and Sue, G. B. (1999) "Intelligent Agent Platform for Procurement" In IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan.

[25] Lee, R.S.T., and Liu, J. N. K.(2004) "iJADE Web-miner: an intelligent agent framework for Internet shopping", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, pp. 461-473.

[26] Zhang, B., Chen, X., Liu, G., and Cai, Q. (2000) "Agent Architecture: A Survey on RoboCup Simulator Teams", In 3rd World Congress on Intelligent Control and Automation.

[27] Ito, N., Shinoda, K., Nakagaw, K., Iwata, K., Du, X., and Ishii, N.(1999) “Action Decision with Incomplete Information: A Temporal Database Approach": In IEEE International Conference on Systems, Man and Cybernetics, Tokyo, Japan.

[28] Kraus, S., and Lehmann, D.(1988) "DIPLOMAT, an Agent in a Multi-Agent Environment: An Overview": In Seventh Annual International Phoenix Conference on Computers and Communications.

[29] Elharrar, D.(2017) "7 Types of Bots: Different ways to deliver value"[online]. Available at: https://chatbotsmagazine.com/7-types-of-bots-8e1846535698 [Accessed 2 February 2018].

[30] Pratt, E. (2017) "Artificial Intelligence and Chatbots in Technical Communication – A Primer" [online]. Available at: http://intelligent-information.blog/wp-content/uploads/2017/09/A-Primer-AI-and-Chatbots-in-Technical-Communication.pdf [Accessed 2 February 2018].

[31] Surmenok, P.(2016) "Chatbot Architecture"[online]. Available at: https://medium.com/@surmenok/chatbot-architecture-496f5bf820ed [Accessed 5 February 2018].

[32] Persiyanov, D.(2017) "Chatbots with Machine Learning: Building Neural Conversational Agents"[online]. Available at: https://blog.statsbot.co/chatbots-machine-learning-e83698b1a91e [Accessed 4 February 2018].

[33] Sansonnet, J. P., Leray, D., & Martin, J. C. (2006). Architecture of a Framework for Generic Assisting Conversational Agents. Intelligent Virtual Agents Lecture Notes in Computer Science, 145–156. http://doi.org/10.1007/11821830_12

[34] Gupta J., Forgionne G. A., Mora M. T.(2006) ‖IntelligentDecision-making Support Systems. Foundations, Applications and Challenges, Springer-Verlag.

[35] Zambonelli, F.(2010) Agent-oriented software engineering [online]. Available at: http://didattica.agentgroup.unimore.it/wiki/images/6/6b/AOSE.pdf [Accessed 31 January 2018].

[36] Butterfield, A. and Ngondi, G. E. (2016) A Dictionary of Computer Science (7th ed.), Oxford University press.

[37] Artale, A.(2008) "FORMAL METHODS"[online]. Available at: http://web.iitd.ac.in/~sumeet/slide3.pdf [Accessed 12 February 2018].

[38] Dam, M.(2015) "Introduction to Temporal Logic"[online]. Available at: http://www.csc.kth.se/~austrin/DD3350/slides/temporal%20logic%20intro.pdf [Accessed 12 February 2018].

[39] De Boer, F. S. (2004) “Methodology for Agent-Oriented Software Design”[online]. Available at: http://www.cs.uu.nl/3apl/projectdescription.html [Accessed 31 January 2018].

[40] Giorgini, P. & Henderson-Sellers(2005) "Agent-Oriented Methodologies:An Introduction"[online]. Available at: http://www.troposproject.eu/files/Henderson01.pdf [Accessed 10 February 2018].

[41] Burmeister, B.(1996). Models and methodology for agent-oriented analysis and design: In K Fischer (ed.), Working Notes of the KI’96 Workshop on Agent-Oriented Programming and Distributed Systems.