

Course Advisory and Results Expert System (CARES): An Implementation of FMI Course Auto-Scheduling Algorithm

Fale Mantim Innocent
Lecturer in FCE Pankshin,
Computer Science Department,
Pankshin, Nigeria

Abdulsalam Ya'u Gital
Lecturer in ATBU Bauchi,
Computer Science Department,
Bauchi, Nigeria

Iliya Nengak Sitlong
Lecturer in FCE Pankshin,
Computer Science Department,
Pankshin, Nigeria

ABSTRACT

Course advisement and results computation are tedious, time-consuming and exhaustive jobs; yet they are necessary as students' performance/success is partly due to them. These jobs engulf cumbersome tasks ranging from course scheduling to guidance and counseling, and then the computation of Grade Point Average (GPA) for students. Improper and untimely advising or computation of results may hinder a student from timely graduation. This paper proposes the FMI course auto-scheduling algorithm which was implemented in an application called "Nixz" – a Course Advisory and Results Expert System (CARES). The inference engine of Nixz, which was programmed using Python, is a hybrid of Rule-Based Reasoning (RBR) and Case-Based Reasoning (CBR). Nixz was built to house both prescriptive and developmental advising models. Nixz reasons through forward chaining. The other programmable components of Nixz were built in C#.NET using Microsoft Visual Studio 2017. The knowledge base of Nixz was built using Microsoft SQL Server 2012.

General Terms

Computer Science, Expert Systems

Keywords

Academic Advising, Course Advisor, Expert System, Knowledge-Based System

1. INTRODUCTION

There is a limit to the performance of humans on any kind of job. But every employer wants to employ speed, accuracy and creativity. Humans wear-off easily – especially when they tend to do the same thing repeatedly. Course Scheduling and Results Computation are tasks that can wear-off any Human Course Advisor (HCA). As they wear-off, their speed, accuracy and creativity deteriorates. On the other hand, a Course Advisory and Results Expert System (CARES) is able to do this and more without losing speed, accuracy and creativity – leaving the HCAs with the other aspects of their job. This bias might explain the uprising of expert systems. Course Scheduling is the guidance of students on course registration while results computation entails calculating students' Grade Point Average (GPA), Cumulative Grade Point Average (CGPA), and grading of students based on their performances.

Expert Systems are programs that emulate or outdo the behavior of human specialists, usually restrained to a particular profession [1]. Expert Systems are domain-specific, but Course Advisement can be global – provided the institutional-based policies are standardized. Nixz, a proposed CARES, is a perfect example of such Expert Systems. Nixz's knowledge base – as proposed – is a combination of Rule-Based Reasoning (RBR)

and Case-Based Reasoning (CBR). An RBR because it will use IF~THEN rules for fresh scenarios after which it will store the scenario and its outcome as new knowledge. A CBR because once it recognizes a similar or same scenario, it will present a similar or same outcome – as the case may be – for such a scenario. This is common with human experts. For example, a Court Judge follows some set of premises before drawing up a verdict on a case. But once a case is similar or same with an already treated one, he presents the verdict of the treated case as a verdict for the new one. This is not only applicable to Court Judges but human experts and non-experts in general.

The prescriptive and developmental models of academic advising are also proposed to be joint-attributes of Nixz. Nixz, using a developmental model of academic advising, will have a module which allows each advisee to perform a 'pilot-registration' before meeting his/her course advisor for confirmation and other form of advisement. On the other hand, Nixz, using a prescriptive model of academic advising, will have a module which allows the course advisor to schedule/register courses for the advisee – with the consent of the advisee.

CARES is of great importance for diverse reasons; multi-objective CARES is not common in institutions – including Federal College of Education, Pankshin (FCEP), Nigeria; the success of this research will aid HCAs and it will also add dynamism / new knowledge to CARES research – considering the approach used in this research (which is pretty different from the ones that shall be reviewed in section 2).

Section 2 of this paper discusses 'related work' while section 3 discusses 'course scheduling: a brief overview'. In section 4, 'the nixz architecture' is discussed. Section 5 presents 'case study' while section 6 covers 'implementation'. Section 7 looks at the 'evaluation' of nixz and section 8 handles 'discussion of results'. Section 9 is 'conclusion'. Section 10 covers 'references'.

2. RELATED WORK

Can there ever be a 'global' expert student advising system that applies to all academic institutions and departments? The variations in academic regulations and policies specific to each academic unit makes this seem unlikely. CARES research has gained great momentum, despite the fact that it has lingered for a while. This quest is partly due to the irregularities in institution-based academic advising and results computation policies. This is believed to have made it impossible for different institutions to use the same CARES – they would rather be built to serve an institution's purpose. Worthy of note is the fact that approach and perspective are constraints to any research. A handful of concerted efforts channeled towards this

research are reported in this paper. The advising models, knowledge base and inference engine will be the only components to be reviewed.

In [2], ‘A Prototype Student Advising Expert System Supported with an Object-Oriented Database’, an expert system called ‘IS-Advisor’ was discussed. This system uses the prescriptive and developmental advising models. Its knowledge base is RBR. The inference engine of ‘IS-Advisor’ was not discussed in [2].

‘Implementation of an Intelligent Course Advisory Expert System’ as discussed in [3] is a Case-Based Course Advisory Expert System. The system’s knowledge base – as reported – is a hybrid of RBR and CBR. The advising model(s) and inference engine used was not discussed in [3].

‘Case Study: A Course Advisor Expert System’ was discussed in [4]. The advising model(s), knowledge base(s) and inference engine(s) used in the implementation of this expert system was not emphatically specified. However, [4] insinuates that the expert system’s knowledge base is a combination of RBR and CBR.

‘An Expert System for Advising Postgraduate Students’ was discussed in [5]. The advising model(s) of Postgraduate Advisor Expert System (PAS) was not specified in [5]. PAS’ knowledge base is an RBR and its inference engine undefined. ‘The Graduate Student Advisor (GSA): An Expert System for SAN Graduate Student Advising’ as discussed in [6] suggestedly uses a developmental advising model. GSA’s knowledge base is an RBR. GSA’s inference engine is undefined but its algorithms suggest that it perceives knowledge through forward chaining.

‘The Graduate Course Advisor: A Multi-Phased Rule-Based Expert System’ as discussed in [7] suggestedly uses a developmental advising model. GCA’s knowledge base is an RBR. GCA’s inference engine reasons through forward chaining. The advising task of GCA is divided into four phases, each of which may apply the inference engine to its own rule-base and invoke other procedures. The GCA was modelled after MYCIN.

‘PACE: a planning advisor on curriculum and enrolment’ was discussed in [8]. PACE uses the prescriptive advising model. Its knowledge base is an RBR. It reasons through forward chaining.

‘Educational Advisor System Implemented by Web-Based Fuzzy Expert Systems’ was discussed in [9]. This system uses the prescriptive advising model. Its knowledge base is an RBR. Its inference engine reasons through forward chaining.

‘Nixz: A Course Advisory and Results Expert System’ differs from a handful of course advisory and results expert systems because it jointly uses an RBR and a CBR. It also implements both the prescriptive and developmental advising models.

Nixz’s knowledge grows as its usage increases. The rules in its RBR can be modified or perhaps, changed with little or no effort – making it cheap to maintain. Nixz’s inference engine – though reasons through forward chaining – is designed to work under the most stringent conditions. The Nixz architecture shall be discussed in section 4.

3. COURSE SCHEDULING: A BRIEF OVERVIEW

Going by either the prescriptive or developmental model of advising, the first ever registration of students capture their personal or bio data, departmental information and admission requirements. These give a course advisor a glimpse about a student. Subsequent registration(s) follow the below procedure:

- i. Check if student’s tuition and other fees have been paid.
- ii. Check student’s previous performance to determine whether or not he/she is qualified to register for the current session/level.
- iii. Determine courses to be registered.
- iv. Maximum and minimum credit limits should be checked before registering a course.
- v. Failed or carryover courses are registered first.
- vi. When a prerequisite course is failed, no dependant courses can be registered.
- vii. Multiple registrations of courses are not allowed.
- viii. Course advisors or students are only allowed to freely add/drop elective courses.

Registration procedures are usually made available to students at every registration period. These procedures/guidelines are usually pasted on notice boards, uploaded to students’ portal and/or included in students’ handbook as the case may be. This is to ensure a healthy registration process and also to establish a level of understanding between the course advisor and the student. Nixz’s Rule-Based (RB) and Case-Based (CB) course scheduling algorithms shall be discussed in section 6.

4. THE NIXZ ARCHITECTURE

As shown in Fig.1, Nixz is based on a 3-tier architecture consisting of a client, application and data tiers. The client tier enables course advisors and students to request for course scheduling recommendations using a web browser via client devices like PCs, mobile and smart phones, and personal digital assistants (PDAs). There are graphical user interfaces (GUIs) available on this tier for interactions between users and Nixz.

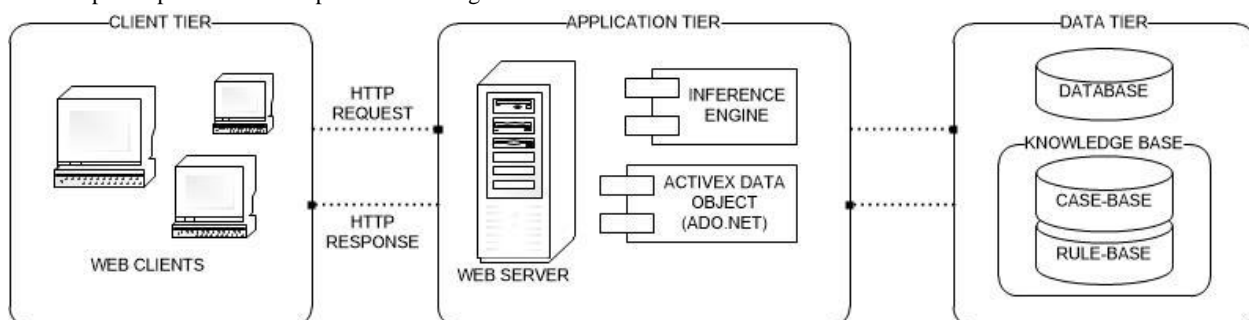


Fig.1: A 3-tier Architecture of Nixz

The application tier consists of a web server, inference engine and ActiveX Data Object (ADO).NET. The web server handles the processing of requests from the client tier. It mediates between the client tier and the data tier. It coordinates the activities of the inference engine and ADO.NET. It also handles computations of results after which it stores it in the database – this later serves as knowledge to the inference engine. The inference engine is the mechanism for decision making. It interacts with the knowledge base and database via ADO.NET and provides course advisors and students with the best course scheduling recommendations. Just as presented in the flowchart in Fig.2, the inference engine accepts new cases, attempts to match them with already treated cases in the case-base. If similar cases are found, the inference engine adopts their solutions for the new cases; otherwise it relies on the rule-base for solutions for the new cases – after which it stores the new cases-solutions pairs in the case-base. ADO.NET is responsible for storing and retrieving information from both database and knowledge base of the data tier.

The data tier comprises of the database and the knowledge base. The database consists of institution-based programs, faculties, departments, courses, students’ results, and etcetera. The knowledge base is constituted by a rule-base and a case-base.

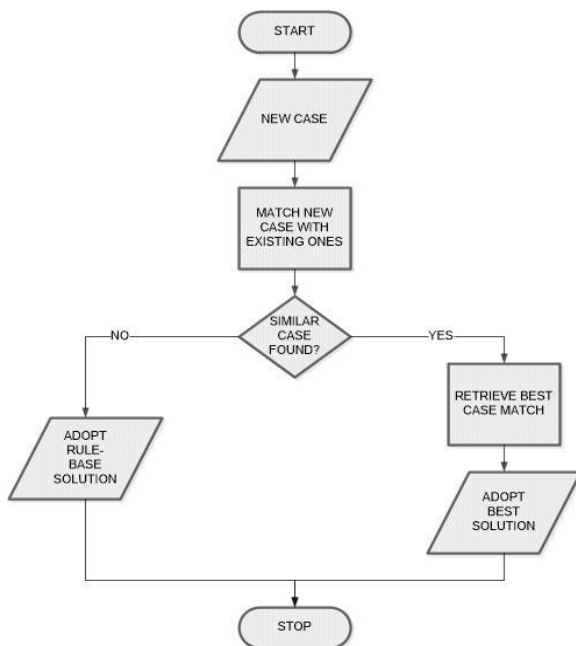


Fig.2: Flowchart of the Inference Engine recommendation on course scheduling

The case-matching mechanism of the inference engine is based on fuzzy logic. The flowchart, as shown in Fig. 2, best describes how the inference engine matches a new case with old ones in order to develop a solution for such a case. The inference engine accepts new case. Attempts to match new case with existing ones. If similar case(s) is/are found, it retrieves and adopts the best possible solution else, it adopts a rule-based solution.

The case-matching mechanism of the inference engine seeks for similarities between a new case and old cases. These similarities are ranked from 0.0 to 1.0. These values are calculated by the formula:

$$\Omega = \frac{\alpha[nc, oc]}{\alpha[nc, oc] + \beta[nc, oc]} \quad (1)$$

Where: Ω =similarity; nc=new case; oc=old case;
 $\alpha[nc,oc]$ =number of courses that match nc and oc; and
 $\beta[nc,oc]$ =number of courses that do not match nc and oc.

Once the similarities between the new case and the old case are calculated, the solution to the one that is closest to 1.0 is adopted as the solution to the new case. A similarity that equals 0.0 means that there are no intersections between the new case and the old case used in the calculation. While a similarity of 1.0 means that there are intersections (but no differences) between the new case and the old case used in the calculation. Any similarity that falls between 0.0 and 1.0 imply that there are both intersections and differences between the new case and the old case used in the calculation.

5. CASE STUDY

Federal College of Education, Pankshin (FCEP) is a tertiary institution located in Plateau State, Nigeria. FCEP has seven faculties/schools and thirty three departments. FCEP has more than ten thousand students. FCEP runs a list of programs including fulltime and part-time degrees (in affiliation with University of Jos, Nigeria), fulltime Nigeria Certificate in Education (NCE), and part-time NCE. Nigerian institutions run one or more of the aforementioned programs as necessitated. Except for a few institution-based policies guiding course registration, the course registration process is standardized in Nigeria. Standards bodies like the National Universities Commission (NUC), National Commission for Colleges of Education (NCCE), etcetera have guidelines for course registration in institutions and conducts as such. But nonetheless, the FCEP course registration procedure has formed the basis for Nixz’s course scheduling algorithm which shall be discussed in section 6.

6. IMPLEMENTATION

Table 1: FMI Course Auto-Scheduling Algorithm

Algorithm AUTOCOURSESCHEDULING(StudentID,C,F,R,E)

INPUT: vector C of registration presets from current level downwards for student, vector F of failed courses for student, vector R of overall distinct registered courses for student, vector E of permissible elective courses for student

OUTPUT: vector X of recommended courses for the current registration

```

if voluntaryWithdrawal(StudentID)==false && withdrawn(StudentID)==false then
  <*-remove overall distinct registered courses from vector C of registration presets for student*->
  for each r in R
    remove r from C
  rof;
  
```

```
<*register failed courses for student*>
for each f in F
  if (creditRegistered+creditOf(f))<= maxCreditAllowed then
    add f to X
    creditRegistered:=creditRegistered+creditOf(f)
  fi;
rof;

<*register courses from vector C of registration presets for student*>
for each c in C
  if (creditRegistered+creditOf(c))<=maxCreditAllowed then
    if courseAlreadyAdded(c)==false && coursePrerequisiteFailed(c)==false then
      add c to X
      creditRegistered:=creditRegistered+creditOf(c)
    fi;
  fi;
rof;

<*register courses from vector E of permissible elective courses for student*>
for each e in E
  if creditRegistered+creditOf(e)<=maxCreditAllowed then
    if courseAlreadyAdded(e)==false && coursePrerequisiteFailed(e)==false then
      add e to X
      creditRegistered:=creditRegistered+creditOf(e)
    fi;
  fi;
rof;

return X;
fi;
end;
```

Nixz's course scheduling algorithm as shown in Table 1 generates a list of prescribed core courses from current level down to the previous level(s) for respective students and stores in vector C. Nixz then generates overall distinct registered courses for such students and stores it in vector R. All the courses in vector R are now removed from vector C – leaving only courses that have never been registered for these students.

Now, considering the allowable number of credits to register, Nixz generates a list of failed courses (vector F) for each student and adds to vector X of recommended courses. Nixz then attempts to add vector C to vector X. If the credits limit is not reached, Nixz generates prescribed electives and allows the respective students to add to vector X

Table 2 shows the results computation algorithm of Nixz. Nixz totals the overall credits registered, finds the overall sum of the product of credit by grade point of each course for each student,

and then performs a division of this sum by credits registered to give the Cumulative Grade Point Average (CGPA).

Table 2: Nixz's Result Computation Algorithm

Algorithm COMPUTERESULT(StudentID,C)

INPUT: vector C of registered courses for current level downwards for student

OUTPUT: Cumulative Grade Point Average CGPA

```
<*compute CGPA*>
CGPA:=0.00

for each c in C
  creditRegistered:=creditRegistered+creditOf(c)
  totalGradePoint:=totalGradePoint+creditOf(c)*gradePointOf(c)

  if creditRegistered <> 0 then
    CGPA:=totalGradePoint/creditRegistered
  fi;
rof;
```

```

if CGPA < 1.00 then
    probate(StudentID)
fi;

return CGPA;
end;
    
```

Fig.3 and Fig.5 show how course scheduling is done by Nixz. Nixz requires that the user inputs an academic session and level, after which it generates a list of prescribed courses to be registered. Fig.4 shows a sample of results computed for

students. This result captures Cumulative Total Credits Registered (CTCR), Cumulative Total Credits Earned (CTCE), Cumulative Total Grade Point (CTGP) and Cumulative Grade Point Average (CGPA).



Fig.3: Course Scheduling for 200L ~ 2015/2016

RESULT SUMMARY							
TCR	TCE	TGP	GPA	CTCR	CTCE	CTGP	CGPA
47	47	120	2.55	86	86	201	2.33

Fig.4: Sample Result generated by Nixz

Courses Registered				
	Title	Code	Load	Score
<input type="checkbox"/>	Computer Programming I	CS201	4	50
<input type="checkbox"/>	Computer Programming II	CS202	3	64
<input type="checkbox"/>	Introduction To Computer Systems	CS203	2	70
<input type="checkbox"/>	Introduction To File Processing	CS204	2	60
<input type="checkbox"/>	Philosophy Of Education	EDU201	2	54
<input type="checkbox"/>	Educational Psychology II	EDU202	2	60
<input type="checkbox"/>	Sociology Of Education	EDU203	2	68
<input type="checkbox"/>	Introduction To Computer Studies	EDU204	2	56
<input type="checkbox"/>	Instructional Strategies	EDU205	2	50
<input type="checkbox"/>	Elements Of Special Education	EDU206	2	65
<input type="checkbox"/>	Phylosophy And Logic	GST102	2	57
<input type="checkbox"/>	PEACE AND CONFLICT RESOLUTION	GST222	2	70
<input type="checkbox"/>	Entrepreneurship Studies	GST223	2	72
<input type="checkbox"/>	Sets, Logic And Algebra	MTH203	3	57
<input type="checkbox"/>	Linear Algebra I	MTH204	3	67
<input type="checkbox"/>	Linear Algebra II	MTH205	2	50
<input type="checkbox"/>	Introduction To Numerical Analysis	MTH209	3	52
<input type="checkbox"/>	Electrical Circuits And Electronics	PHY202	3	56
<input type="checkbox"/>	Inference	STA231	4	58
Total Credit Registered =				47

Fig.5: Sample of scheduled courses by Nixz

7. EVALUATION

COURSE SCHEDULING

1	The predictions of Nixz are accurate?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
2	Nixz requires less input to make predictions?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
3	Nixz does not demand for irrelevant information?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

RESULTS COMPUTATION

4	The results computed by Nixz are accurate?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
5	Nixz requires less input to compute results?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
6	Nixz probation mechanism is accurate?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
7	Nixz withdraws students on the right basis – that is multiple-probation?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
8	Nixz predicts that students who pay tuition but did not apply for deferment of academic sessions and did not register their courses have voluntarily withdrawn from the institution. And also predicts that students who did not pay their tuition have voluntarily withdrawn from the institution?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

SYSTEM SECURITY

9	Nixz does not allow unauthorized access?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
10	Nixz does not keep a user logged in for more than six hours?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

COST EFFECTIVENESS & RELIABILITY

11	Nixz drastically cuts down course advising time?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
12	Nixz knowledge grows as the system usage grows?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
13	Nixz does not breakdown or hang?				
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

Fig.6: Questionnaire for the evaluation of Nixz

In order to ensure proper evaluation of Nixz, a beta-test was performed by a sample of ten experts (human course advisors) and ten non-experts (students) from FCEP. A questionnaire (see Fig.6) was administered to both experts and non-experts to evaluate their experience of Nixz. Each question in the questionnaire has options based on Rensis Linkert's scale (Strongly Disagree=1; Disagree=2; Neutral=3; Agree=4; Strongly Agree=5).

8. DISCUSSION OF RESULTS

The findings of Nixz's evaluation (as described in Table 3) is a comparison of means – with its dependent and independent variables coded (CS=Course Scheduling; RC=Results Computation; SS=System Security; CER=Cost Effectiveness and Reliability; HCA=Human Course Advisors; STD=Students).

Table 3: Results of Nixz's Evaluation

GROUP	CS	RC	SS	CER
HCA	4.52	4.94	4.20	4.36
STD	4.33	4.50	4.40	4.26
AVERAGE OF MEANS	4.43	4.72	4.30	4.31

The result of this analysis shows the different HCA and STD ratings of Nixz. The average rating shows that the course scheduling mechanism's accuracy of Nixz is rated 88.6% while its results computation mechanism is rated at 94.4% accuracy. Nixz is 86% secure. It is also 86.2% cost-effective and reliable.

9. CONCLUSION

If a system can be built for FCEP – to accommodate its list of programs, faculties/schools, departments, and the number of students – then such a system can be extended to other institutions with FCEP-like scenarios; Nixz is also workable in institutions that run master's degree, post-graduate diploma and diploma.

Nixz is built for institutions of higher learning. There is no fear of its workability in Nigeria. Nixz is currently used for course scheduling and results computation at FCEP. But despite the acceptability that this system might have earned, there may still be room for improvement. The future scope of this research is with regards to knowledge acquisition and method of inference. Further research on Nixz will be a

modification of its inference engine; an improvement on this will enlarge the precincts of its prescription and recommendation mechanisms. In order to achieve this, a multi-phased architecture will be adopted. The rule-base design is also a very important and interesting area of research as its success will ensure more accurate prescriptions and recommendations.

10. REFERENCES

- [1] J. S. Robert, *Intelligent Systems: Principles, Paradigms and Pragmatics*, Sudbury, Massachusetts: Jones & Bartlett Publishers, 2009.
- [2] M. A. A. Ahmar, "A Prototype Student Advising Expert System Supported with an Object-Oriented Database," *International Journal of Advanced Computer Science and Applications (IJACSA)*, Special Issue on Artificial Intelligence, pp. 100-105, 2011.
- [3] Daramola, O. Emebo, I. Afolabi and C. Ayo, "Implementation of an Intelligent Course Advisory Expert System," *International Journal of Advanced Research in Artificial Intelligence (IJARAI)*, vol. 3, no. 5, 2014.
- [4] O. Noran, "Case Study: A Course Advisor Expert System," in *AI 2003: Advances in Artificial Intelligence*, Berlin Heidelberg, Springer, 2003, pp. 1014-1026.
- [5] Al-Ghamdi, S. Al-Ghuribi, A. Fadel, F. Al-Aswadi and T. Al-Ruhaili, "An Expert System for Advising Postgraduate Students," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 3, no. 3, pp. 4529-4532, 2012.
- [6] J. Zhang, "The Graduate Student Advisor (GSA): An Expert System for SAN Graduate Student Advising," *School of Engineering & Applied Science*, Miami University, Oxford, 1992.
- [7] M. G. Valtorta, B. T. Smith and D. W. Loveland, "The Graduate Course Advisor: A Multi-Phase Rule-Based Expert System," in *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, 1984.
- [8] H. Guandi, K.-H. Lim and W.-Y. Yeong, "PACE: a planning advisor on curriculum and enrollment," in *Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences*, 1995.
- [9] H. G. Mahdi and R. Vahid, "Educational Advisor System Implemented by Web-Based Fuzzy Expert Systems," *Journal of Software Engineering and Applications*, pp. 500-507, 2012.