Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 31 (2014) 22 – 31

Procedia Computer Science

2nd International Conference on Information Technology and Quantitative Management, ITQM 2014

# Rule-based expert systems for supporting university students

Gökhan Engin[a], Burak Aksoyer[b], Melike Avdagic[b], Damla Bozanlı[b], Umutcan Hanay[b], Deniz Maden[b], Gurdal Ertek[b*]

[a] Oracle Turkey, Istanbul, 34398, Turkey
[b]Faculty of Engineering and Natural Sciences, Sabancı University, Istanbul, 34956, Turkey

**Abstract**

There are more than 15 million college students in the US alone. Academic advising for courses and scholarships is typically performed by human advisors, bringing an immense managerial workload to faculty members, as well as other staff at universities. This paper reports and discusses the development of two educational expert systems at a private international university. The first expert system is a course advising system which recommends courses to undergraduate students. The second system suggests scholarships to undergraduate students based on their eligibility. While there have been reported systems for course advising, the literature does not seem to contain any references to expert systems for scholarship recommendation and eligibility checking. Therefore the scholarship recommender that we developed is first of its kind. Both systems have been implemented and tested using Oracle Policy Automation (OPA) software.

*Keywords:* rule-based systems; expert systems; recommender systems; decision support systems; higher education; university education.

## 1. Introduction

### *1.1. Rule-Based Expert Systems*

Rule-based expert systems have the ability to emulate the decision making ability of human experts. They are designed to solve problems as humans do, by exploiting encoded human knowledge or expertise[1]. This knowledge can be extracted and acquired directly through interaction with humans, as well as from printed and electronic resources such as books, magazines and websites. The extracted knowledge forms the knowledge

---

* Corresponding author. Tel.: +90-216-483-9568 ; fax: +90-216-483-9550.
E-mail address: ertekg@sabanciuniv.edu.

base of the rule-based system. The other major component of rule-based systems draws conclusions from this knowledge, and is referred to as the inference engine. The conclusions and suggestions offered by the rule-based system satisfy users' needs for expertise within the chosen domain.

Rule-based systems are designed to solve problems in a selected domain. Every domain has its own knowledge and reasoning humans, which can be emulated and even replaced through automated rule-based systems. Many domains contain a large amount of knowledge that can be captured fully only through an information system, since humans may not access or immediately retrieve fully the needed information. There are many advantages of rule-based expert systems: They decrease costs since they reduce the need for human experts; they are permanent; they can be used for different knowledge systems, which increases functionality; they increase reliability since they minimize errors that humans are prone to; and if designed by multiple experts, can increase confidence. Finally, they lack human emotions, which are sources of mistakes in human based systems[1]. The advantages of rule-based expert systems are multifold and they can considerably facilitate human life for the better. In this paper, we present and describe two rule-based recommender systems projects, both in the domain of university education.

## *1.2. Oracle Policy Automation (OPA)*

The software used in the reported projects is Oracle Policy Automation (OPA). OPA is capable of reading rules conveniently from spreadsheet and word-processor file formats, where the rules are written in a format akin to natural language. OPA automatically forms a problem solving or decision making application based on the acquired rule base. In OPA, the rules can be written in more than 25 languages, including Turkish, the language selected for the expert system in our second case study. Users codify the information or knowledge via Microsoft Word and/or Microsoft Excel in their native (natural) language and form rules without having to know any programming languages (such as Java or C#) or expert system languages (such as CLIPS). This makes the testing and maintenance of, as well as the changes to the rules easier. If the rules are detailed and need frequent modifications, and/or if there will be auditing and the rules need to be transparent, OPA is especially suitable and convenient. After the program is executed and a suggestion is presented by the program, the program also shows why and how that suggestion is selected. This way, the rules can be checked and the reasons behind the suggestions can be provided to users.

Writing and running an OPA program has five stages. First one is to gather the information and data for the rules that will be implemented. Second step is to write the rules using Microsoft Excel and/or Word. Afterwards, the rule base is checked for language mistakes, rule mistakes and other errors that might have occurred. After the rule base is finalized, the users start answering the related questions to obtain a suggestion. In the first project, the suggestion is the list of courses that can be taken by a student, given the courses he has taken so far. In the second project, the suggestion is the eligibility of (a student for) a specific scholarship. After all the questions are answered, the rule-based expert system presents its suggestion. For example, in the second project, the resulting output is in the form "The student/user is eligible for the Sabancı Vakfı Scholarship" or "The student/user is not eligible for the Sabancı Vakfı Scholarship", corresponding to the suggestions of applying and not applying to the Sabancı Vakfi Scholarship, respectively. The user can see the reason why the program gave the result it gave by clicking the underlined result. If the user is eligible for the scholarship, clicking the underlined hyperlinked text will show all the criteria satisfied by the user. If the user is not eligible for the scholarship, then the underlined hyperlinked text will state which criteria was not satisfied. This part of OPA is very useful for auditing reasons because it clearly states the reasons for the results. In our application this is highly relevant because the school systems, entrance exams and the rules for the eligibility change frequently.

### *1.3. Expert Systems for University Education*

There are more than 6000 colleges and more than 15 million college students in the US alone[2], and academic advising is typically performed by human advisors[3]. Engaging with students is becoming more complex and student-demand driven[4], bringing an immense managerial workload to faculty members, as well as other staff at universities. There is also an information overload on the student side, as the information that needs to be interpreted, clarified, and captured is showing an ever increasing trend[4]. Expert systems developed for university education can serve a variety of goals, including advising, tutoring, instruction, and evaluation. We will discuss the studies in literature under two groups, namely expert systems for course advising and expert systems for other education-related goals. Within the first group we will also discuss course advising systems that are built using methodologies other than expert systems.

Dunstan[3], Nambiar and Dutta[5], Laghari & Khuwaja[6], and Al Ahmar[7] report expert systems for advising undergraduate students on course selection. Al-Ghamdi *et al.*[8] describe a system for advising postgraduate students and presents a detailed review of expert systems for course selection. Khanna *et al.*[9] also provide an extensive review, besides listing the main areas where expert systems can be used in education. Onyeka *et al.*[10] propose combining rule-based reasoning with case based reasoning in course advising expert systems. Goodarzi and Rafe[11] incorporate fuzzy reasoning into its advisory expert system. Koutrika *et al.*[12] develop a flexible system for generating advisory expert systems. Parameswaran *et al.*[2] present CourseRank, a course recommendation and planning system developed at Stanford University. The authors report that more than 170 universities throughout the world have adopted CourseRank, making the software in the most widely used course advising system in the world.

Besides expert systems for course advising, the literature contains a multitude of studies that describe course advising systems built using other methodologies. Studies of Vialardi *et al.*[13], Taha[14], Cho and Kang[15], and Park *et al.*[16] are examples where course recommendation is based on data mining. Among these three studies, Vialardi *et al.*[13] reports using real world data from a Spanish university; Taha[14] uses collaborative filtering; Cho and Kang[15] uses hybrid filtering. Sobecki[17] uses nature inspired methods, again for course recommendation.

Besides course recommendation, expert systems have been developed for other purposes, as well. Buche and Querrec[18] describe an expert system for helping human learners in virtual reality environments. Hwang *et al.*[19] introduce an expert system that works as instructor for improving problem solving skills of students. Jaryani *et al.*[20] present an expert system that provides customized learning for each student based on his/her background and realities. The fuzzy expert system in the work of Van Hecke[21] identifies the personal profile of the students as one of the three types. Grupe[22] presents a web based expert system for selecting academic major, where the student user is directed to one of the most suitable six majors from among 60. Deorah *et al.*[23] also recommend academic majors. Fong and Biuk-Aghai[24] describe a system that recommends the university that matches the student's profile. The system developed by Aslam and Khan[25] is one for recommending faculties to students. Finally, Cline and Brewster[26] present an expert system for automatically evaluating student concept maps.

## 2. Case 1: Supporting Course Selection Decisions

### *2.1. Motivation*

For this project our aim was to create a rule-based system for Manufacturing Systems Engineering Program students at Sabanci University to prepare their schedules using Oracle Policy Automation (OPA). When preparing the rule base, we investigated rules related to prerequisite courses, student's GPA (Grade Point Average), which courses open which semester in, which area the student is specializing in, and which courses the student has readily registered to.

## *2.2. Project Progress*

In order to reach our aim we collected data from the university's database of courses, namely BannerWeb. First of all we listed all courses in MS Excel and we connected them with their pre-requisites. This way the obtained a lists of the courses and the relations between courses. Furthermore, we represented this data in the yEd graph visualization software. It is possible to assign colors to nodes in yEd. This way, connection between courses and three main course types as area electives, core electives and required courses became clearly distinguishable (Figure 1).

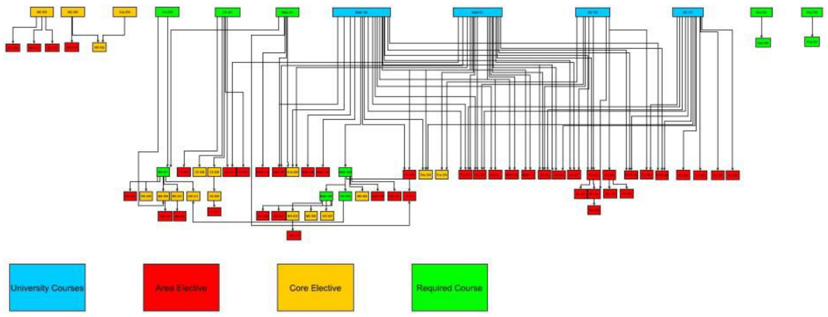


Fig. 1. Visualization of co- and pre-requisite relationship between courses

We used the logical connectors "and" and "or" to establish logical connection between a course and its pre-requisites. If a user should take all pre-requisites of a course, "and" is the operator to identify relationships. On the other hand, if one of pre-requisites of a course is sufficient to take that course, "or" is the operator to describe the relationship. Also, through the "exists" command, system associates one entity with another entity. We established the course definition as an excel table with four columns, which are CourseID, CourseName, and CourseCredit and CourseCategory. Another Excel document used by OPA is the student database, which receives student number as input. By this single input, OPA matches student number with student name from the student list. Also, OPA saves all historical data of a user for the next session with the user. This way, a user can easily observe her/his historical movement on OPA as a user. For instance, a student can easily follow his/her credit and course level at past semesters. In addition to this, OPA calculates total credits for all sub-groups of courses is required, core and area electives. And then, OPA gives the user total credit by summation of all units. We finally planned and executed various scenarios and test cases for testing the developed export system.

In addition to the mentioned rules, several other types of rules have been defined:

*Global Controls and Determinations:* These are rules that globally determine student and credit attributes, such as the following:

```
eligibility check is completed if
        total credit is known and
        student name is known
the student is sophomore if
        total credit > 47 and
        total credit < 81
```

*Individual Course Eligibility Checks:* These are rules that determine student eligibility for each course such as the following:

```
the student is eligible for ENS204 if
the student has taken NS101
and
either
        the student has taken MATH101 or
        the student has taken MATH102
```

*Course technical rules:* These are technical rules that tell which courses are taken, such as the following:

```
the student has taken CS201
        Exists(all courses of the student, course id = “CS201”)
the student has taken ENS208
        Exists(all courses of the student, course id = “ENS208”)
the student has taken ENS491
        Exists(all courses of the student, course id = “ENS491”)
the student has taken ENS492
        Exists(all courses of the student, course id = “ENS492”)
```

## 2.3. Results

The developed rule-based expert system meets the user with a welcome screen. Next screen takes as input the student number and in order to match student name and record action logs of the user.
Third screen is designed for taking course list from the user (Figure 2). The user chooses his / her previously taken courses from a list, where the list is read from a source file by OPA. Once the user submits the courses taken, the lists of courses that s/he can take is listed (Figure 3).

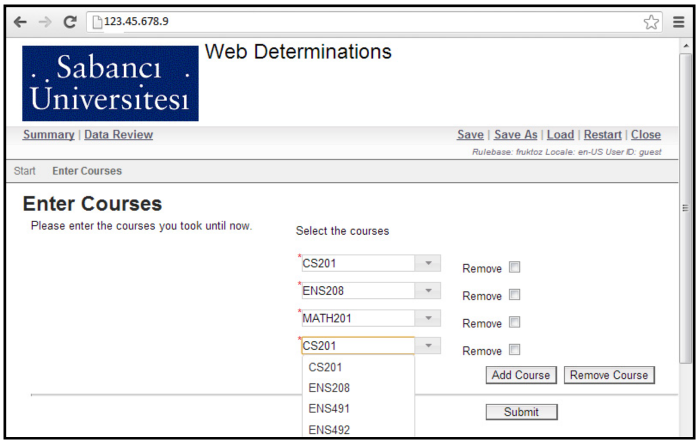
If a user wants to know the reason why s/he is not eligible for a course, the user should click “Why?” button on the results screen to reach that information, and learn the reason (Figure 4).

## 3. Case 2: Supporting Scholarship Decisions

### 3.1. Motivation

The goal of the second project was also to assist university students by providing recommendations. After discussions in the early stages of our project, we decided to implement a scholarship advisor that will present, with a student’s criteria, which scholarship s/he is eligible for. There are many institutions that offer scholarships to university students, however every foundation’s criteria and budget is different. Our aim was to match universities students with the scholarships that they are eligible for. This can help students in easily finding what they are looking for and it can decrease a task that would take weeks to minutes.

We used OPA again for this project. In our case, the rules would be based on the students’ GPA, their target university, their income, their skills, and other relevant criteria. OPA asks multiple choice questions that are related with the criteria of the scholarships. From the home screen, the student clicks on the scholarship and then answers the related questions, and finally the results screen shows if s/he is eligible for the scholarship or not.


123.45.678.9

Web Determinations

Summary | Data Review

Save | Save As | Load | Restart | Close

Rulebase: fruktoz Locale: en-US User ID: guest

Start Enter Courses

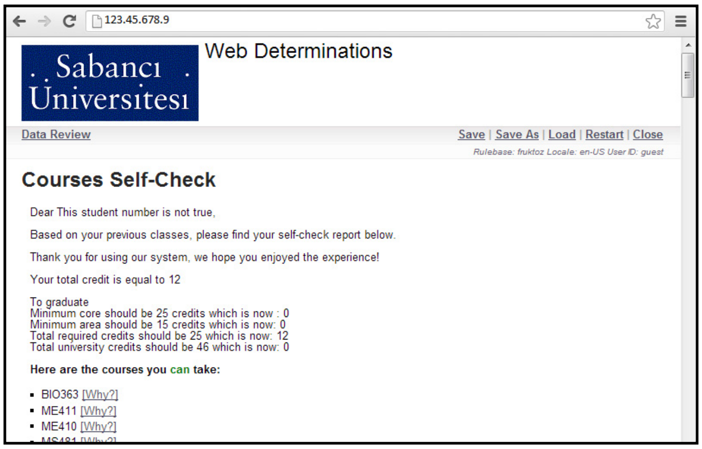
**Enter Courses**

Please enter the courses you took until now.

Select the courses

| | |
|---|---|
| CS201 | Remove |
| ENS208 | Remove |
| MATH201 | Remove |
| CS201 | Remove |

Fig. 2. Screen where the user (student, advisor, staff member) enters the courses taken so far


123.45.678.9

Web Determinations

Data Review

Save | Save As | Load | Restart | Close

Rulebase: fruktoz Locale: en-US User ID: guest

**Courses Self-Check**

Dear This student number is not true,

Based on your previous classes, please find your self-check report below.

Thank you for using our system, we hope you enjoyed the experience!

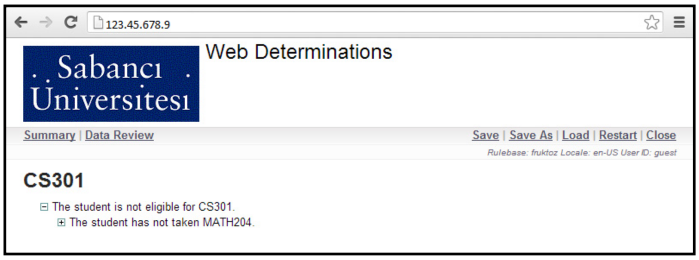
Your total credit is equal to 12

To graduate
Minimum core should be 25 credits which is now : 0
Minimum area should be 15 credits which is now: 0
Total required credits should be 25 which is now: 12
Total university credits should be 46 which is now: 0

**Here are the courses you can take:**

- BIO363 [Why?]
- ME411 [Why?]
- ME410 [Why?]

Fig. 3. Screen where the courses that the student can take are listed


123.45.678.9

Web Determinations

Summary | Data Review

Save | Save As | Load | Restart | Close

Rulebase: fruktoz Locale: en-US User ID: guest

**CS301**

⊟ The student is not eligible for CS301.
  ⊞ The student has not taken MATH204.

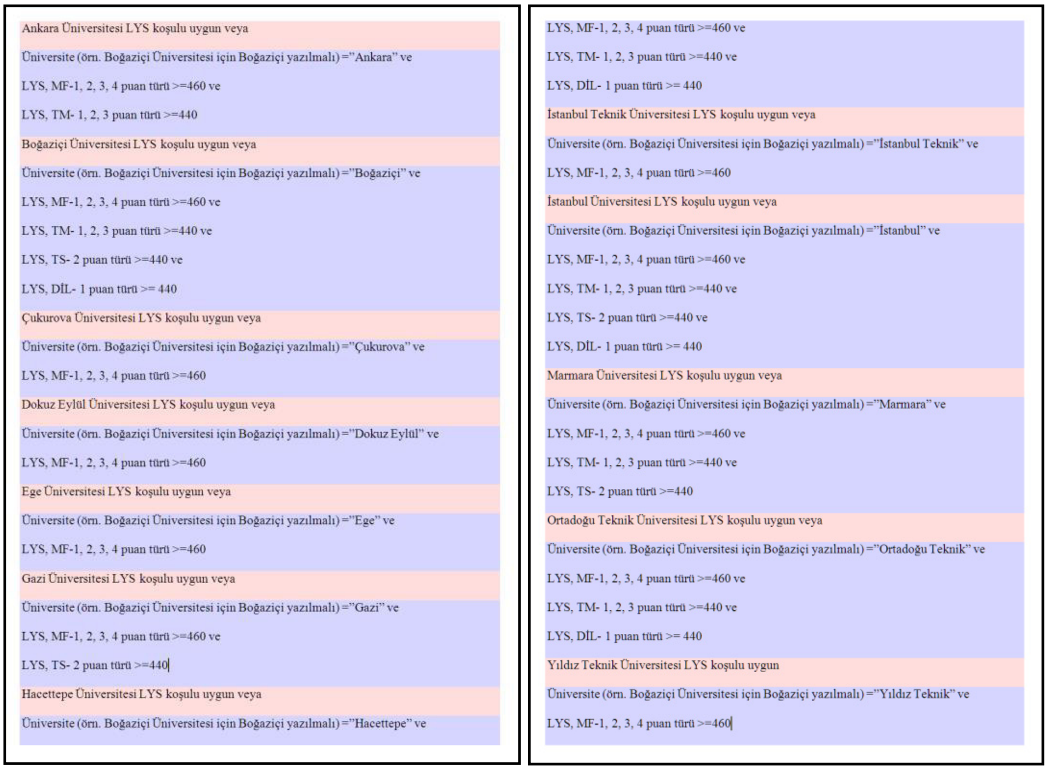
Fig. 4. The screen where the rationale for the suggestion is provided

## *3.2. Project Progress*

To realize the project, we firstly gathered the data of all the scholarships in Turkey and their criteria. Finding all the rules wasn’t easy since some of the scholarships did not have a website but had just a phone number. Criteria for some of the scholarships were not available on the websites since the application period was over. After scanning through scholarships, we found approximately 200 scholarships, half of which were only reachable by phone and didn’t provide a website. We eliminated those scholarships from further consideration. From among the other 100 scholarships, only 51 had the criteria publicly available on their websites. From that point on, we deducted the rules of eligibility for each scholarship and separated every one of them in an Excel sheet so that while writing the rules for OPA, it would be easier and faster.

Furthermore, after gathering all the data we parsed their requirements and pre-requisites. For one institution, there may be an income requirement, while for another, meeting just a GPA threshold is sufficient, and thus we divided all the information into subcategories. To make it more organized, every scholarship’s criteria is listed in an Excel file so that every rule can be seen and plugged into OPA easily. With this system the students’ search and information gathering time can decrease significantly, and they can filter through significant amount of information and work to achieve on their goals. Moreover, we believe students would not be the only people who would appreciate this program. Also, foundations can receive applications from students who are more likely to be selected, and consequently their workload can be reduced, as well.

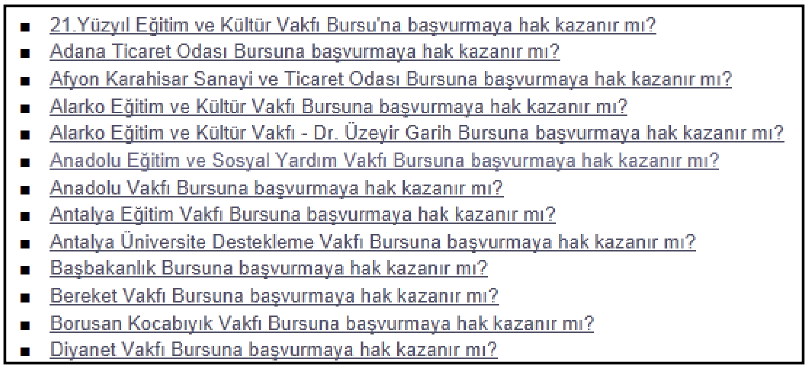
The second stage of the project consisted of writing all the rules in OPA. Figure 5 illustrates rules for a particular scholarship. This rule writing was done for each of the 51 scholarships.



Ankara Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Ankara” ve
LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TM- 1, 2, 3 puan türü >=440
Boğaziçi Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Boğaziçi” ve
LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TM- 1, 2, 3 puan türü >=440 ve
LYS, TS- 2 puan türü >=440 ve
LYS, DİL- 1 puan türü >= 440
Çukurova Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Çukurova” ve
LYS, MF-1, 2, 3, 4 puan türü >=460
Dokuz Eylül Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Dokuz Eylül” ve
LYS, MF-1, 2, 3, 4 puan türü >=460
Ege Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Ege” ve
LYS, MF-1, 2, 3, 4 puan türü >=460
Gazi Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Gazi” ve
LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TS- 2 puan türü >=440
Hacettepe Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Hacettepe” ve

LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TM- 1, 2, 3 puan türü >=440 ve
LYS, DİL- 1 puan türü >= 440
İstanbul Teknik Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”İstanbul Teknik” ve
LYS, MF-1, 2, 3, 4 puan türü >=460
İstanbul Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”İstanbul” ve
LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TM- 1, 2, 3 puan türü >=440 ve
LYS, TS- 2 puan türü >=440 ve
LYS, DİL- 1 puan türü >= 440
Marmara Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Marmara” ve
LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TM- 1, 2, 3 puan türü >=440 ve
LYS, TS- 2 puan türü >=440
Ortadoğu Teknik Üniversitesi LYS koşulu uygun veya
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Ortadoğu Teknik” ve
LYS, MF-1, 2, 3, 4 puan türü >=460 ve
LYS, TM- 1, 2, 3 puan türü >=440 ve
LYS, DİL- 1 puan türü >= 440
Yıldız Teknik Üniversitesi LYS koşulu uygun
Üniversite (örn. Boğaziçi Üniversitesi için Boğaziçi yazılmalı) =”Yıldız Teknik” ve
LYS, MF-1, 2, 3, 4 puan türü >=460

Fig. 5. Rules for a particular scholarship



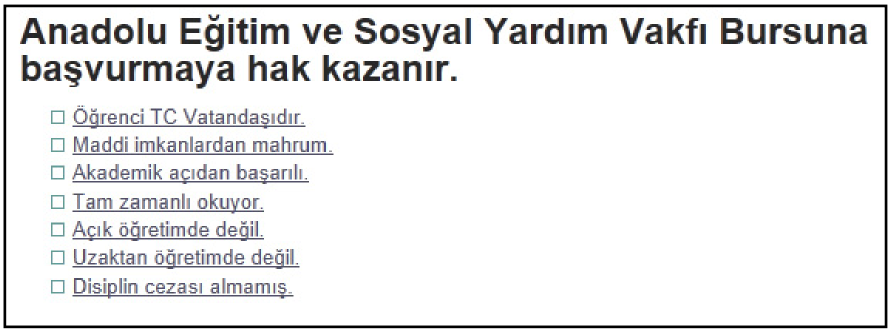
- 21.Yüzyıl Eğitim ve Kültür Vakfı Bursu'na başvurmaya hak kazanır mı?
- Adana Ticaret Odası Bursuna başvurmaya hak kazanır mı?
- Afyon Karahisar Sanayi ve Ticaret Odası Bursuna başvurmaya hak kazanır mı?
- Alarko Eğitim ve Kültür Vakfı Bursuna başvurmaya hak kazanır mı?
- Alarko Eğitim ve Kültür Vakfı - Dr. Üzeyir Garih Bursuna başvurmaya hak kazanır mı?
- Anadolu Eğitim ve Sosyal Yardım Vakfı Bursuna başvurmaya hak kazanır mı?
- Anadolu Vakfı Bursuna başvurmaya hak kazanır mı?
- Antalya Eğitim Vakfı Bursuna başvurmaya hak kazanır mı?
- Antalya Üniversite Destekleme Vakfı Bursuna başvurmaya hak kazanır mı?
- Başbakanlık Bursuna başvurmaya hak kazanır mı?
- Bereket Vakfı Bursuna başvurmaya hak kazanır mı?
- Borusan Kocabıyık Vakfı Bursuna başvurmaya hak kazanır mı?
- Diyanet Vakfı Bursuna başvurmaya hak kazanır mı?

Fig. 6. The first screen that the user sees in Case Study 2

### 3.3. Results

The lists of possible scholarships (in this program there are 51 scholarships) appear on the user's first interface. The user starts the program by clicking on the scholarship she/he wants to test the eligibility for the first. There are some common rules required for most of the scholarships, the question is asked only once and used in other scholarships as well. Examples for these common rules are the school student wants to attend or is attending, students major, and is whether the student is a Turkish citizen or not (most frequent criterion).

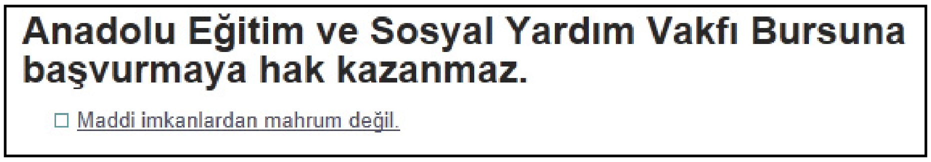
In Figure 6, the first interface that the user sees is presented. Here in our example, let us assume that the student clicks the link "Anadolu Eğitim ve Sosyal Yardım Vakfı Bursuna başvurmaya hak kazanır mı?" ("Is the student eligible for the Anadolu Eğitim ve Sosyal Yardım Vakfı scholarship?") and starts answering the questions in order. Questions appear successively as they are answered in a way that preserves eligibility. If the answer violates eligibility, then the expert system automatically stops and prompts that "The student is not eligible for this scholarship". For the specific scholarship mentioned, the correct sequence of answers would be (yes, yes, yes, yes, no, no, no), and if this sequence is entered the reason page would look like Figure 7. If, for example, the second question is answered no instead of yes, the reason page would look like Figure 8.



**Anadolu Eğitim ve Sosyal Yardım Vakfı Bursuna başvurmaya hak kazanır.**

- ☐ Öğrenci TC Vatandaşıdır.
- ☐ Maddi imkanlardan mahrum.
- ☐ Akademik açıdan başarılı.
- ☐ Tam zamanlı okuyor.
- ☐ Açık öğretimde değil.
- ☐ Uzaktan öğretimde değil.
- ☐ Disiplin cezası almamış.

Fig. 7. Result screen displayed when the student is eligible, with the rationale for the suggestion (eligible, apply) is explained


Anadolu Eğitim ve Sosyal Yardım Vakfı Bursuna başvurmaya hak kazanmaz.

☐ Maddi imkanlardan mahrum değil.

Fig. 8. Result screen displayed when the student is *not* eligible, with the rationale for the suggestion (not eligible, do not apply) is explained

### *3.4. Discussions*

The biggest problem this rule-based system presented was with the questions where variables were used. Rather than the system asking whether someone's GPA is greater than 3.5, we decided it would be better if the program asked what the GPA was and then decided if it satisfies the rules. The problem arose when for example we wanted to know the students major. It would have been impractical to ask the user to answer questions like "Are you a Business major?" and if the answer is no asking "Are you a Industrial Engineering major?" and so on. So again we decided to use the variables. But since the program is written in Turkish, the user could enter the major using capital letter, small letter, Turkish letters or English letters. This makes a huge number of possibilities. This is where we decided to search for a pull-down menu for these variables.

The student using this automation doesn't have to answer all the questions for the 51 scholarships; by clicking on the few scholarships they want to apply they can answer the related questions. The system also doesn't require the user to answer the same question more than once even if the question's answer is required for different scholarships. For example the citizenship criteria is one of the popular ones, so in the first eligibility test the user answers it and the program remembers the answer and does not ask the same question for the other scholarships.

Our first thought was for the student to write down all his or her information and then all the available scholarships would have been presented. Unfortunately due to the fact that every scholarship has different rules and the importance of the answers change in different scholarships, that kind of system would not have worked. In the end, we have assembled the information and the available software and completed the case study successfully.

## 4. Conclusions

This paper reported and discussed the development of two educational expert systems: The first expert system is that course advising system which recommends courses to undergraduate students. The second system suggests scholarships to undergraduate students based on their eligibility. While there have been reported systems for course advising, the literature does not seem to contain any references to expert systems for scholarship recommendation and eligible checking. Therefore, the scholarship recommender that we developed is the first such system in the literature and can provide a frame of reference for following studies. Both systems have been implemented and tested using Oracle Policy Automation (OPA) software, a convenient integrated development environment for expert systems development. Future stages of the project can involve the deployment of the developed expert systems on the university intranet and/or the internet.

**Acknowledgements**

The authors thank Mete Sevinç for creating the visualization of co- and pre-requisite relationships between the courses.

## References

1. Giarratano JG. *Expert Systems Principles and Programming*. USA: PWS Publishing; 2002.
2. Parameswaran A, Venetis P, Garcia-Molina H. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS),* 2011, **29**(4), 20.
3. Dunstan N. ET: an Enrolment Tool to Generate Expert Systems for University Courses, In: *Expert Systems,* Petrica Vizureanu (Ed.), ISBN: 978-953-307-032-2, InTech, DOI: 10.5772/7071. 2010. Available from: http://www.intechopen.com/books/expert-systems/et-an-enrolment-tool-to-generate-expert-systems-for-university-courses or the short link http://bit.ly/19nnbp6. Accessed on November 14, 2013.
4. *Oracle Policy Automation for Higher Education.* Oracle Data Sheet. Available under short link http://bit.ly/1boStNO. Accessed on November 13, 2013.
5. Nambiar AN, Dutta AK. Expert system for student advising using JESS. In: 2010 *International Conference on Educational and Information Technology (ICEIT)*, *IEEE.* 2010, **1**, p. V1-312.
6. Laghari, MS, Khuwaja GA. Electrical engineering department advising for course planning. In: *Global Engineering Education Conference (EDUCON), 2012 IEEE*. 2012, p. 1-6.
7. Al Ahmar MA. A Prototype Student Advising Expert System Supported with an Object-Oriented Database. *(IJACSA) Int J Adv Comp Sci Appl*. 2011. Available under http://www.ijacsa.thesai.org. Accessed on November 13, 2013.
8. Al-Ghamdi A, Al-Ghuribi S, Fadel A, AL-Ruhaili FAAT. An Expert System for Advising Postgraduate Students. *(IJCSIT) Int J Comp Sci and Info Tech,* **3**(3) , 2012, 4529-4532.
9. Khanna S, Kaushik A, Barnela M. Expert systems advances in education. In: *Proceedings of the National Conference on Computational Instrumentation NCCI-2010. CSIO,* Chandigarh, India. 2010, p. 109-112.
10. Onyeka E, Olawande D, Charles A. CAES: A model of an RBR-CBR course advisory expert system. In: *2010 International Conference on Information Society (i-Society),*. IEEE. 2010, p. 37-42.
11. Goodarzi MH, Rafe V. Educational Advisor System Implemented by Web-Based Fuzzy Expert Systems. *Journal of Software Engineering and Application (JSEA),* 2012. 5(2).
12. Koutrika G, Bercovitz B, Ikeda R, Kaliszan F, Liou H, Garcia-Molina H. Flexible recommendations for course planning. In: *IEEE 25th International Conference on Data Engineering, 2009 (ICDE'09).* 2009, p. 1467-1470.
13. Vialardi C, Bravo J, Shafti L, Ortigosa A. Recommendation in higher education using data mining techniques. *Group on Educational Data Mining,* 2009, p. 190-199. Available from: http://eric.ed.gov/?id=ED539088. Accessed on March 28, 2014.
14. Taha K. Automatic academic advisor. In: *Collaborative Computing: 8th International Conference on Networking, Applications and Worksharing (CollaborateCom)*. 2012, p. 262-268.
15. Cho J, Kang EY. Personalized Curriculum Recommender System Based on Hybrid Filtering. In: *Advances in Web-Based Learning–ICWL 2010*, 2010, p. 62-71. Springer Berlin Heidelberg.
16. Park DH, Kim HK, Choi IY, Kim JK. A literature review and classification of recommender systems research. *Expert Syst Appl,* 2012, **39**(11), 10059-10072.
17. Sobecki J. Comparison of nature inspired algorithms applied in student courses recommendation. In: *Computational Collective Intelligence. Technologies and Applications,* 2012, p. 278-287. Springer Berlin Heidelberg.
18. Buche C, Querrec R. An expert system manipulating knowledge to help human learners into virtual environment. *Expert Syst Appl,* 2011, **38**(7), 8446-8457.
19. Hwang GJ, Chen CY, Tsai, PS, Tsai CC. An expert system for improving web-based problem-solving ability of students. *Expert Syst Appl,* 2011, **38**(7), 8664-8672.
20. Jaryani F, Sahibudin S, Ibrahim S, Rahman NA, Daruis R. Intelligent reflective e-portfolio framework based on artificial intelligent Expert systems techniques. In: *3rd International Conference on*. *Computer Research and Development (ICCRD), IEEE,* 2011, **2**, p. 214-216.
21. Van Hecke T. Fuzzy Expert System to Characterize Students. *PRIMUS,* 2011, **21**(7), 651-658.
22. Grupe FH. An Internet-based expert system for selecting an academic major: www.MyMajors.com. *Internet High Educ,* 2002, **5**(4), 333-344.
23. Deorah S, Sridharan S, Goel S. SAES-expert system for advising academic major. In: *2nd International Advanced Computing Conference (IACC), IEEE,* 2010, p. 331-336.
24. Fong S, Biuk-Aghai RP. An Automated University Admission Recommender System for Secondary School Students. In: *The 6th International Conference on Information Technology and Applications*. 2009.
25. Aslam MZ, Khan AR. A Proposed Decision Support System/Expert System for Guiding Fresh Students in Selecting a Faculty in Gomal University, Pakistan. *Ind Eng Letters,* 2011, **1**(4), 33-40. Available under http://www.iiste.org. Accessed on November 13, 2013.
26. Cline BE, Brewster CC, Fell RD. A rule-based system for automatically evaluating student concept maps. *Expert Syst Appl,* 2010, **37**(3), 2282-2291.