

Expert system for supporting the students english language essay

[Linguistics](#), [English](#)



Ana Tănăsescu Petroleum and Gas University of Ploiești, București Avenue,
No. 39, Ploiești, Romania e-mail: atanasescuro@yahoo. com

Abstract

In this paper it is presented an expert system that computes the transferable credits number that was accumulated by a student and its grades averages. This system aim is to facilitate the personnel activity from a faculty secretariat. In the paper first part, domain knowledge modelling, they are described the phases that were followed in order to build the Student ontology, a frame - based ontology. In the paper second part, expert system design, they are presented the two main components of the knowledge base that the author designed them using Jess Expert System Shell, namely: fact base and rule base. Key words: students' record ontology, students' record expert system, rule base, fact base, Jess Expert System ShellJEL

Classification: C63, C81, I23

Introduction

The main activity of a faculty secretariat is a student centred activity. This implies: the students registration, the record of students' grades, transferable credits corresponding to the courses from the curricula, scholarships that students can benefit, as well as different taxes that they are required to pay. This paper aims to present an expert system that computes the transferable credits that were accumulated by a student at the end of an academic year and his average, if the student has passed all the exams. This expert system role is to support and facilitate the activities that are realized by the personnel from a secretariat and, therefore, the data

about the students are taken from a frame - based ontology. The two main phases that constituted the base for the expert system building are further described: the domain knowledge modelling and the expert system design.

Domain Knowledge Modelling

The knowledge modelling from a specialty domain represents a knowledge transfer process from domain human experts and/or other knowledge sources to expert systems. Some of the most known techniques for knowledge modelling are [1],[2],[3]: knowledge engineering techniques, ontologies, modelling languages, multi-perspective modelling techniques. The author has chosen ontologies as a knowledge modelling technique from academic management domain, in particular, the record activities management domain of students' grades and transferable credits that are accumulated by them during an academic year. The ontology term was defined as a " explicit specification of a conceptualization"[4],[5]. It describes the basic concepts in a domain and the relationships between them [6],[7]. Among the methodologies that can be used at ontologies building, presented in the specialty literature [8],[9],[10], the author has used the Uschold and King methodology that contains four phases: the ontology purpose identifying, the ontology building, the ontology evaluation and the ontology documentation. The Protégé ontology editor was chosen from the numerous development environments and tools that have appeared lately, in order to effectively realize the Student ontology. This is an ontology editor and knowledge-base framework that has been evolving for over a decade [11],[12],[13],[14]. Protégé editor provides the following advantages [15]: is

based on Java, can run on a variety of platforms, supports customized user-interface extensions, incorporates the Open Knowledge Base Connectivity knowledge model, allows export of its ontologies in different formats and has been used by hundreds of individuals and research groups. A list of questions that knowledge base developed based on the ontology should be able to answer must be elaborated in order to realize the first phase of the ontology building process, namely ontology purpose identifying. Some of the questions that the ontology presented in this paper must answer are further enumerated: What are the specializations that can be attended at Economic Sciences Faculty; What courses do the curricula of these specializations contain and what are their characteristics (number of lecture hours, number of application hours, credits number, assessment type etc.)? Who are the persons that teach these courses? Who are the students that attend the specializations from this faculty? Thus, it can be said that the ontology purpose is to support and to facilitate the record activities of students' grades and transferable credits accumulated by them, activities that are realized by the personnel from the faculty secretariat. The ontology building phase includes the following activities too: reusing existing ontologies, enumeration important concepts and terms in the ontology, defining the classes and class hierarchy, defining the properties of the classes and their facets. The Student ontology presented in this paper was created by the author from scratch, none of the ontology described in the specialty literature or existing on the market being used. The author has used the oldest and the most used knowledge elicitation manual method, namely the interview, in order to determine and characterize the key concepts and

terms of the ontology. For this purpose, the author interviewed both the personnel from the faculty secretariat and the personnel from the faculty management that is responsible with the teaching activities management. The author has, also, used for the knowledge acquisition the direct observation of the experts during their work, another knowledge elicitation manual method. Some of the questions from the interview are further presented: What are the courses that are studied at the faculty specializations? Who are the persons that teach those courses and from what departments do they belong to? Who are the faculty students, as well as what grades and credits did they accumulate until that moment? How is elaborated a curriculum, what information does it contain and how are allocated the courses' codes in a curriculum? What is European Credit Transfer and Accumulation System and what is its purpose? Thus, the author could identify the following terms: teacher, student, specialization, group, course, course code, assessment type, hours number, credits number that is allocated to a course, grade and credits that were accumulated by a student etc. The author have chosen among the previously identified terms only those that describe the fundamental concepts existing in the students' record activity management domain and has defined the ontology classes and class hierarchy. As it can be seen in figure 1, the ontology contains 9 classes: 1 abstract class - Person class and 8 concrete classes - Teacher class and Student class (subclasses of Person abstract class), as well as Group class, Specialization class, Course class, Department class, Assessment class and Date class. However, the classes do not provide enough information to answer to the questions defined in the first phase of

the ontology design. Therefore, after the author defined the classes, he had to describe their internal structure, namely he defined the classes properties and their facets.

Person

Date

Assessment

Student

is-ais-ahashasis-taken-by

Department

belongs_to_department

Teacher

has_members

Course

Group

is_taught_byteachesis_taken_atis_composed_ofbelongs_to_groupis_includedi
s_made_ofis_studied_athas_alocated

Specialization

Fig. 1. The ontology class hierarchy. Four of the nine ontology classes are further presented: Person class, Student class, Course class and Assessment class. They were chosen because have a significant role in the ontology.

Person class is an abstract class that has two direct, concrete subclasses that inherits its properties (Teacher class and Student class). The five

properties of the Person class are (fig. 2): 4 of string type (address, first_name, last_name and npc) and 1 of instance type (date_of_birth). Fig. 2. Person class. Student class is a concrete class, subclass of the Person abstract class. It has 9 properties (fig. 3): 5 inherited from the superclass and 4 proper properties. These are: student_ID, an integer property; initial, a string property; financing_state, a symbol property; belongs_to_group, an instance property that can store only instances of the Group class and that has as an inverse slot the is_composed_of slot that belongs to the Group class. Fig. 3. Student class. Course class is a concrete class that has 10 properties (fig. 4): 2 string properties: course_code and name; 5 integer properties: credit_points, no_hours_lecture, no_hours_application, semester and year_study; 1 symbol property: assessment_type; 2 instance properties: is_studied_at (an instance of the Specialization class) and is_taught_by (an instance of the Teacher class). Both properties have inverse slots.

Assessment class (fig. 5) is a concrete class that contains information about student grades. The instances of this class are described by 5 properties: grade, an integer property; qualificative, a symbol property; course, date and student, instance properties. Fig. 4. Course class. Fig. 5. Assessment class.

Student ontology evaluation, the penultimate phase of the ontology building process, was achieved by instances creation (population the knowledge base with knowledge) and using them as input data in the expert system, as well as by querying these instances with the help of the persons interviewed in the enumeration phase of the important concepts for the ontology. The last ontology building phase is represented by the ontology documentation, namely the explanation of the terms used in the ontology so that they will be

understood by the persons that are not familiar with the ontology domain too.

Expert system design

The author has used Java Expert System Shell (Jess) in order to design the expert system for supporting the students' record activity. Jess is an expert system shell and a script language, centred on Java, providing to its users an automatic access to application programming interfaces for networking, graphics, database connection[16],[17],[18]. Jess is, also, compatible with the Protégé ontology editor because Jess inference engine can be attached to the editor through its Jesstab plug-in. Jess expert system shell is an ideal tool for adding the production rule technology in the software systems so that the method used by the author to represent the knowledge is the production rule method[19]. In order to define the rules from the expert system for supporting the record activity of the grades and transferable credits that were accumulated by students, the author has interviewed the personnel from the faculty secretariat and the personnel responsible with the teaching activity management. Actually, the interview phase coincides with one of the ontology building process phase, namely the enumeration stage of the important concepts and terms for the ontology. The addressed questions were, essentially, about the credit transfer and accumulation system. Thus, the author could acquire the following knowledge: the credits number that can be accumulated in a semester and in an academic year, the minimum credits number that is required to a student in order to pass in the second year, respectively, in third year. Some of the rules modelled based on

this knowledge are further presented: Rule 1 IF credits_no_year1 \geq 60 and qualificative_PhE1 = A and qualificative_PhE2 = A THEN The Student has passed all the exams of the first year and can attend the second year. Rule 2 IF credits_no_year1 \geq 40 and credits_no_year1 < 60 and credits_no_year2 \geq 60 THEN The Student has not passed all the exams of the first two years of study, but he can attend the third year of study. Since the designed expert system is a rule-based expert system (knowledge is represented by production rules), the knowledge base contains two components: fact base and rule base. The expert system fact base contains three facts categories: shadow facts, unordered facts and ordered facts. Shadow facts are facts that ensure the connection with Java objects[20]. In the expert system presented in this paper, the shadow facts are represented by the facts obtained through the import of instances belonging to six classes from Student ontology. Classes instances import is realized by a mapping process, using a Protégé ontology editor plug-in called Jesstab plug-in[21]. Unordered facts are similar to the records from the relational database tables or to the objects from the object-oriented languages and slots correspond to the fields from the table that stores the record, respectively, to the properties of the class that includes the object[22]. Two unordered facts categories were used in the expert systems: credits facts that store information about student, courses where the student has passed the exams and their corresponding credits, as well as the classbook facts that contain information about student, courses with the exam passed by the student and their grades. Credits facts template is: (deftemplate credits (slot last_name)(slot first_name)(slot initial)(slot coursecode)(slot course)(slot year)(slot

creditpoints))Ordered facts " are simply Jess lists where the first element acts as a sort of category for the facts"[23]. These were used in order to activate the rules from the expert system. Thus, in the fact base are added the following ordered facts: student fact that contains information about the student that is searched by the user; qualificative facts corresponding to the qualificatives received at Physical Education Course; creditsyear facts that include the credits number that were accumulated in an year; integral facts that indicate the year where the student passes all the exams; avgyear facts that store the grades average for the years where the student has passed all the exams; overallavg fact that contains the grades overall average of the student after he has completed the three years of study. Rule base, the knowledge base second component, represents a set of rules that are activated and run depending on the fact base content. The author has divided and organized the rules in modules, in order to prevent the conflict between them, to control their execution and to facilitate their retrieval. The rule base of this expert system contains 46 rules and 4 functions organized in 9 modules (fig. 6). These modules are: mapping module, read module, identifyingstudent module, creditsqualificatives module, creditscalculation module, displaycredits module, classbook module, avgcalculation module and displayavg module. Mapping module is used to import the instances of Student ontology classes from Protégé ontology editor through the mapping process. Thus, the shadow facts that represent the input data of the expert system are added in the application fact base. Read module contains 3 functions that are called so that the user specifies the data about the student whose credits and grades average will be computed. Identifyingstudent

module displays information about the searched student and, also, adds a student ordered fact in the working memory of the expert system.

Creditsqualificatives module includes 2 rules that are activated based on the Course class instances (mapped in shadow facts in the first module) and based on student ordered fact (added in the fact base in identifyingstudent module). When these rules run, in the fact base are added two more fact categories: credits unordered facts and qualificatives ordered facts.

Creditscalculation module computes the credits total number that was accumulated by the analyzed student in each year of study and inserts, in the fact base, one creditsyear ordered fact with credits total number for each attended year of study. Displaycredits module displays the credits total number accumulated by a student in each year of study. If the student has obtained the required credits total number in a year, then an integral ordered fact with information about that year of study is added in the working memory.

User

User interface

MAPPING

MODULE

Protégé

READ

MODULE

IDENTIFYINGSTUDENT MODULE

adding ruleimportedshadow factsglobal variables with information about the student

CREDITSQUALIFICATIVES MODULE

rules that generate credits and qualificative factsimportedshadow

factsimportatetestudent ordered fact

CREDITSCALCULATION MODULE

credits rulecredits unordered facts

DISPLAYCREDITS MODULE

display rulescreditsyear ordered factsqualificative orderedfactsinformation

about the accumulated credits

CLASSBOOK MODULE

generatefacts ruleintegral ordered factsclassbook unordered

factsimportedshadow facts

DISPLAYAVG MODULE

display rulesavgyear ordered factsoverallavg ordered factinformation about the student grades average per years and about the student overall average

AVGCALCULATION MODULE

rules for averages computing

Fig. 6. Interconnectivity of rule base modules. Displaycredits module contains 32 rules. The author has chosen to present

further the code of two rules. These rules were, previously, defined in

pseudo-code.(defrule displaycredits:: display1_1(MAIN:: creditsyear 1 ? cn1&

(eq ? cn1 60))(MAIN:: qualificative " Physical Education I" A| ME)(MAIN::

qualificative " Physical Education II" A| ME)(not(MAIN:: creditsyear 2 ?))

(not(MAIN:: creditsyear 3 ?))

=>

(assert(MAIN:: integral 1))(printout t " The student is in the first year of study and has accumulated a credits number=? cp1 ", passing all his exams."

crlf))(defrule displaycredits:: display2_11(MAIN:: creditsyear 1 ? cn1&

(and(>= ? cn1 40)(< ? cn1 60))(MAIN:: creditsyear 2 ? cn2&(>= ? cn2 60))

(not(MAIN:: creditsyear 3 ?))

=>

(assert(MAIN:: integral 2))(printout t " The student is in the second year of study")(printout t " and he has accumulated " ? cp1 " credits number in first

year of study and " ? cp2 " credits number in the second year of study. He

has not passed all the exams of the first year of study, but he can attend the

third year of study." crlf))Classbook module inserts in the fact base the

classbook unordered facts with information about grades taken by the analyzed student at the passed courses only for the years of study where he has passed all the exams. Avgcalculation module computes based on the classbook unordered facts existing in the fact base the grades average for the years where the student has passed all the exams. If the student has ended the third year of study and has passed all the exams from the three years of study, the system computes his overall average. Displayavg module contains 7 rules that display the student grades averages for the years where he has passed all the exams and the overall average, if it is necessary.

Conclusions

This paper describes an expert system for supporting the students' record activity from the Economic Sciences Faculty. Its aim is to facilitate the personnel activity from the faculty secretariat and to reduce their work volume. This system can be used, also, in other faculties by adapting the rules from expert system to the requirements about the credits total number existing in the curricula of those faculties. In this paper is, also, presented an ontology whose instances are imported as an input data in the expert system. This ontology can be easily develop by other knowledge engineers that are interested in the field, can be imported in other decision support systems from educational domain and can train the users less familiar with the academic management domain.

Bibliografie

Abdullah, M. S., Benest, I., Evans, A., Kimble, C., Knowledge Modelling Techniques for Developing Knowledge Management Systems, 3rd European Conference on Knowledge Management, Dublin, Ireland, September 2002, ISBN 0-9540488-6-5, pp. 15-25. Casely-Hayford, L., A comparative analysis of methodologies, tools and languages used for building Ontologies, available at <http://epubs.cclrc.ac.uk/bitstream/894/OntologyReport.pdf>. [accessed on 10th of March 2013]. Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A., Methodologies, tools and languages for building ontologies. Where is their meeting point?, Data & Knowledge Engineering, vol. 46, nr. 1, July 2003, pp. 41-64. Fernández López, M., Overview Of Methodologies For Building Ontologies, Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, August 2, 1999, available at <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/4-fernandez.pdf> [accessed on 10th of March 2013]. Friedman-Hill, E., Jess in Action. Rule-Based Systems in Java, Manning Publications Co, 2003. Friedman-Hill, E., Jess 7.0p1. The Rule Engine for the Java™ Platform, Sandia National Laboratories, 2006. Genari, J. H. et al, The Evolution of Protégé: An Environment for Knowledge-Based Systems Development, 2002, available at http://bmir.stanford.edu/file_asset/index.php/52/BMIR-2002-0943.pdf. [accessed on 10th of March 2013]. Gruber, T., What is an ontology, available at <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> [accessed on 10th of March 2013]. Gruber, T., Towards principles for the design of ontologies used for knowledge sharing, International Workshop on Formal Ontology, March 1993, Padova, Italy. Makhfi, P., Introduction to Knowledge <https://assignbuster.com/expert-system-for-supporting-the-students-english-language-essay/>

Modeling, available at http://www.makhfi.com/KCM_intro.htm [accessed on 10th of March 2013]. Noy, N. F., McGuinness, D. L., *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, 2001, available at http://protege.stanford.edu/publications/ontology_development/ontology101.pdf. [accessed on 10th of March 2013]. Oprea, M., Tănăsescu, A., Tehnici de modelare a cunoașterii în sistemele bazate pe cunoștințe, în *Managementul cunoașterii în universitatea modernă*, coord.: Bodea C. și Andone I., Editura ASE București, 2007, ISBN 978-973-594-953-2, pp. 223-242. Sachs, E., *Getting Started with Protégé-Frames*, June 2006, available at http://protege.stanford.edu/doc/tutorial/get_started/get-started.pdf [accessed on 10th of March 2013]. Tănăsescu, A., Bodea, C. N., Pătrașcu, A., *An Environmental Ontology Design*, International Conference Science and Technology in the Context of Sustainable Development, Petroleum-Gas University of Ploiești, Ploiești, November 6-7, 2008, *Buletinul Universității Petrol-Gaze din Ploiești, Seria Științe Economice*, vol. LX, nr. 5B, 2008, ISSN 1224-6832, pp. 181-188. Tănăsescu, A., Pătrașcu, A., Oprea, C., *Environmental Protection Domain Modelling*, Proceedings of the 11th IBIMA Conference, Innovation and Knowledge Management in Twin Track Economies, Cairo, Egypt, 4-6 January 2009, ISBN 978-0-9821489-0-7, pp. 1164-1170. Zaharia, M., Cârstea, C., Sălăgean, L., *Inteligența artificială și sistemele expert în asistarea deciziilor economice*, Editura Economică, București, 2003. <http://protege.stanford.edu>. <http://www.jessrules.com/>. <http://www.ida.liu.se/~her/JessTab/>.

Sistem expert pentru asistarea activității de evidență a studenților unei facultăți

Rezumat

În această lucrare se prezintă un sistem expert care calculează numărul de credite transferabile acumulate de un student și mediile corespunzătoare acestuia. Scopul acestui sistem este să faciliteze activitatea personalului din cadrul secretariatului unei facultăți. În prima parte a lucrării, modelarea cunoștințelor din domeniu, sunt descrise etapele parcurse pentru a construi ontologia Student, ontologie bazată pe cadre. În cea de-a doua parte a lucrării, proiectarea sistemului expert, sunt prezentate cele două componente principale ale bazei de cunoștințe pe care autorul le-a proiectat folosind generatorul de sisteme expert Jess, și anume: baza de fapte și baza de reguli.