# Automating Academic Advising and Course Schedule Planning with a CLIPS Expert System

### **Edwin Rudolph and Adel Abunawass**

Department of Computer Science, University of West Georgia, Carrollton, Georgia, USA

Abstract – A significant aspect of advising students is the process of reviewing the students' academic course history, determining progress toward completion of degree requirements, verifying satisfaction of various pre-requisites, and identifying potential course schedule plans for future enrollment. To assist advisors in accurately and efficiently completing these tasks and to enable more time for qualitative and personalized discussion with individual students, we developed a simple expert system that automates those tasks. Related to advising is the administrative problem of determining what courses should be offered to meet students' future needs. To aid and support decision making, the advising expert system is designed to process data for multiple students simultaneously and project the number of students needing to take certain courses in the future. This poster discusses the general design and implementation of this system using the CLIPS expert system tool.

**Keywords:** Expert system; decision support system; academic advising; CLIPS

# **1** Motivation

As part of efforts to improve and streamline advising of students in our department, we began looking for ways to ensure students would receive consistent advice on their academic progress as well as on issues relating to various university administrative polices and procedures, while also ensuring that each student had the opportunity to receive substantive and personalized advisement on issues such as personal and career goals and research and job opportunities. The challenge with the former was that the routine and largely mechanical advising tasks were dominating the advisement process, and the fact that several faculty members participated in this process also made it difficult to ensure that students received consistent information.

Our approach to addressing this challenge was to implement a two-tier mandatory advising process where the routine aspects of advising (reviewing progress toward a degree, consulting on what courses to take for upcoming semesters, discussion about administrative policies and procedures, etc.) are centralized with a professional staff advisor, with whom the student must first meet, and following this, the student meets with an assigned faculty advisor where academic, career, research, and personal issues may be discussed. This process, which all students must complete during each term of enrollment, achieved both the goal of ensuring students receive consistent advice about degree requirements and various policies and procedures, while improving the quality and content of the consultation with a faculty member. However, to do this in an effective manner, we needed to find a way to automate the analysis of the student's progress to further ensure consistency and accuracy as well as enable the staff advisor to efficiently meet with a large number of students over a relatively short time period (typically a one to two month period preceding the course registration period).

# 2 **Requirements**

We identified three functional requirements for a system that would help us to automate key advising tasks:

- To aid in reviewing a student's progress towards degree completion, list all requirements for the degree and indicate for each whether or not the student has satisfied the requirement. For satisfied requirements, indicate the course(s) used to satisfy those requirements. Since different sets of requirements may exist depending on when the student matriculated, the system must allow for multiple requirement sets and allow the appropriate set to be selected on demand.
- To aid in planning a student's future course schedule, list important program courses the student needs to take, the future term(s) the department expects to offer those courses, and whether or not the student is eligible to take each course based on whether or not the student has satisfied the pre-requisite(s).
- To assist in planning what courses should be offered in future terms by projecting students' needs, list important program courses and for each indicate how many students are eligible to take the course (based on having satisfied applicable pre-requisites).

Important non-functional requirements included the ability to easily code new and/or modified degree requirements as requirements change over time and to be able to handle input and output in a simple text-based format.

## **3** Implementation

An initial prototype system was developed as a webbased application using procedural code in PHP. However, it quickly became obvious that this approach was both tedious to code and difficult to maintain. This initial effort did prove beneficial in that it highlighted the fact that the system was primarily based upon a set of *if-then* rules, which led us to rethink our implementation approach in terms of a rule-based expert system. We chose the CLIPS expert system shell for several reasons: low-cost (public domain), fast (shell implemented in C), a straightforward syntax, and the ability use and parse simple text input and output.

#### 3.1 Overview

The expert system is forward chaining, and uses input facts about the student's academic history (courses taken), the requirements of the student's program of study, and facts about future planned course offerings to produce conclusions (output facts) about what program requirements have or have not been satisfied, and what courses a student may be able to take in the future and when those courses are expected to be offered. The rules that produce those conclusions consist of: program requirement rules that match courses the student has taken with individual program requirements; course pre-requisite rules that match courses the student has taken with pre-requisites needed to enroll in courses that the student has not taken; and ancillary rules that handle issues such as grade replacements for retaking a course, manual overrides to enable course substitutions, and output conversion to enable reporting and parsing of results.

The following figure illustrates the basic components of the expert system as described above:



#### 3.2 Degree Program Requirements Progress Report

To produce the list of all requirements for the degree along with an indication of whether or not a student has satisfied the requirement, the expert system works as follows. First, individual input facts for each course the student has taken along with the grade earned are asserted. Then, facts that describe the individual degree program requirements (e.g. "*two elective courses are required from the following*...") are asserted. For program requirements where there is a single specific course required, the requirement fact is coded to match the course number that would appear in the student's academic history facts. Then, a single rule is used to automatically match all such simple requirements. For more complex requirements where one or more courses are used to determine if the requirement is met, a specific rule is defined to perform the match. Once a match occurs between course(s) from the student's academic history and a program requirement, the requirement fact is modified to indicate that the requirement was met and which course(s) were used to meet it.

#### 3.3 Course Enrollment Eligibility

To produce a list of courses a student has not taken but is eligible to take based on courses already completed (or in progress), the system uses the student's academic history facts (courses taken) to activate pre-requisite satisfaction rules. For example: "*CS 201 requires completion of CS 102 with a grade of C or higher*". This rule would fire and generate a new eligibility fact for CS 201 if the student's academic history contained a fact for CS 102, the grade for that course was a C or better, and the academic history did not already contain a fact for CS 201.

To aid the student in schedule planning, additional input facts are asserted for projected course offerings, indicating that the department expects to offer a course during a given term (e.g. "CS 201 will be offered in Spring 2016"). These facts are then matched with the eligibility facts; resulting in a list of courses the student can take in the future along with the term(s) the department expects to offer those courses.

#### 3.4 Course Offering Needs Projection

To aid in projecting how many students may need to take a certain course, the system uses the same functionality described above for course enrollment eligibility. Instead of only considering a single student, data for all students is asserted (using student IDs to differentiate). Eligibility facts are asserted for each student and these are then counted to produce a projected number of students who may need to take certain courses in the future. For this use case, the program requirement rules for analyzing students' degree progress are not utilized and are disabled so as to minimize processing time.

## **4 Results**

Since 2007, the system has been successfully used in support of over 3,000 advisement sessions and in the planning of course offerings. The CLIPS implementation has enabled us to easily update the system over time to support changes to degree requirements and course pre-requisites (while maintaining older ones), as well as, adapt the analysis engine for use with different user interfaces and reporting needs.