SPILC: An Expert Student Advisor

D. R. Read
Lamar University
Beaumont, Texas

1. Introduction

The Lamar University Computer Science Department serves about 350 undergraduate C.S. majors, and 70 graduate majors. B.S. degrees are offered in Computer Science and Computer and Information Science, and an M.S. degree is offered in Computer Science. In addition, the Computer Science Department plays a strong service role, offering approximately sixteen service course sections per long semester. The department has eight regular full-time faculty members, including the Department Chairman and the Undergraduate Advisor*, and from three to seven part-time faculty members.

Due to the small number of regular faculty members and the resulting very heavy teaching loads, undergraduate advising has become a difficult problem for the department. There is a one-week early registration period and a three-day regular registration period once each semester. The Undergraduate Advisor's regular teaching load of two classes, 6 - 8 semester hours, per semester, together with the large number of majors and small number of regular faculty, cause long queues and short tempers during these advising periods. The situation is aggravated by the fact that entering freshmen are rarely accompanied by adequate documentation containing the facts necessary for proper counselling. There has been no good method of obtaining necessary facts and documenting both the information provided by the student and the resulting advice offered by the counsellors.

Since the requirements for entering the C.S. program are fairly straightforward, and the first two semesters for entering students are reasonably uniform, an expert system that would advise the entering student as to an appropriate schedule appeared to provide the ideal solution to both the shortage in advising personnel, as well as the information gathering and documentation problems. This paper describes the development of such an expert system: SPILC (Student Prompter Involving Limited Communication) written using CLIPS.

2. Goals

The goals of this project were as follows:

1. To evaluate CLIPS for possible inclusion into the Lamar University computer science curriculum,
2. To develop a usable expert system for advising entering freshmen computer science majors,
3. To use the expert advisor as a prototype for a much larger and more sophisticated program for advising and tracking all computer science majors, from entry through graduation.

The evaluation of CLIPS as an expert system tool for use in the classroom had been intended in any case, and that fact, in addition to those features listed in 3., below, encouraged its selection for the expert advisor.

3. Choice of Hardware Platform and Language

Due to the availability of PCs for both development and application of the expert system, it was decided to implement the system for that environment.

Language choice was simplified by the fact that there were only two candidates. Among other factors, the following criteria were used in deciding which candidate to use for the expert advisor:

    Backward chaining support,
    Forward chaining support,
    I/O capability,
    Simplicity and ease of use,
    Low cost,
    Number of copies available,
    Integrated editor.

CLIPS was chosen as the implementation language for this project due mainly to its apparent simplicity and consistency of syntax, the fact that forward chaining was considered to be sufficient for a simple rule-based system, and the department had access to as many copies as it needed for use during advising periods. Since CLIPS was also being considered for possible use in several upper level computer science courses, it was felt that this project would provide an ideal test to determine how easily and quickly it could be learned and used effectively.

4. Architecture of the Expert System

The model chosen was that of a small search space with reliable knowledge and fairly reliable data (1:89-126). While the domain knowledge is very reliable, data provided by the student, as indicated below, can be suspect. Both data and knowledge are reasonably constant over time, and computational resources were considered adequate for the problem.

4.1. Knowledge Acquisition

Expert knowledge was gained from three sources: (i) the Undergraduate Advisor for the Computer Science Department who, due to her very difficult schedule, was limited to a brief (three-page) written description of the typical questions, answers, and decisions that take place during the advising of an entering freshman; (ii) the author's several years experience in advising undergraduates and participating in curriculum development and modification; and (iii) the university undergraduate catalog.

4.2. Domain Knowledge

In order to major in computer science, a first semester student must have a combined score of at least 850 on the SAT (or equivalent ACT), or rank in the top one third of his/her graduating class. A student who has prior credit from another university or college must satisfy those requirements, as well as have an overall gradepoint average of at least 2.3 on all college-level work. After a student is accepted, a departmental "recommended program of study", a standardized degree plan, and the class schedule form the basis for scheduling advice.

The advisor must also consider university policy in such areas as: (i) maximum course load allowed, (ii) a requirement regarding continuous registration for freshman English until credit for six semester hours has been earned, and (iii) a requirement that a freshman must register for physical activity each long semester until he/she has completed four such courses.

Course prerequisite information must be available, as well as information regarding continually evolving general education requirements.

4.3. Student Specific Facts

During a consultation, a considerable amount of information must be collected from each student. Much of the time no official

documentation of the information received from a student is available until well after the registration period has concluded. Often the documentation, when it arrives, is found to be in disagreement with the information supplied by the student during registration. A permanent record of the student-supplied information is desired for both advising purposes as well as for comparison against official documentation. This student-supplied information includes such items as: SAT scores; TASP scores; rank in graduating class; most advanced mathematics course taken successfully; computer science course (and language used) taken successfully; age of student; whether the student has a part-time (or full-time) job, and if so, how many hours per week it requires; and the number of semester hours the student desires to schedule. Some of this student information, such as TASP scores, the highest level mathematics course taken, or rank in class, are required only conditionally.

The decision was made to have the program include the student-supplied data in a hardcopy statement, similar to the following example, to be signed by the student:

<div style="text-align:center">SPILC   March 23, 1990</div>

NAME:	Able, Albert A
SSNUM:	'555-55-5555'
SAT math score:	450
SAT verbal score:	450
1st semester at LU:	yes

Trigonometry or higher passed in HS: yes
Passed a High School C.S. course: yes

To the best of my knowledge, the above information is true. I realize that if any of the above is found to be false, I can be excluded from the Lamar University C.S. Department's degree program.

SIGNED:_____

Recommended Courses:

    C.S. 1411
    Mth 1345
    Eng 131
    Hist 231
    pega 224

If the program determines that the student does not meet the requirements for entering the program, a similar form is printed, indicating the problem and suggesting appropriate action.

5. Design of the Program

A partitioning of the knowledge base was undertaken to simplify both development and debugging, as well as future extension of SPILC. The initial categories for partitioning the rule base were as follows:

1. Rules which controlled the input of permanent student record information, such as name, social security number, SAT scores, etc.;
2. Rules that controlled the input of student scheduling information, such as number of hours desired and number of hours the student works in a part-time job per week;
3. Scheduling rules, which included most of the domain knowledge for the problem;
4. Output rules for printing the acknowledgment of responsibility and the student's recommended schedule.

The facts were partitioned in a similar fashion, but were further subdivided into control facts, student record facts, or scheduling facts.

This partitioning, though convenient, was not necessary for a problem of this small magnitude. It was considered desirable, however, for the purpose of significant future development of the expert system.

6. Future Plans

The prototype is to be field tested during the registration period for the Fall 1990 semester. It will then be modified, as appropriate, to improve the interface and to correct any errors or deficiencies detected at that time. It will be extended to maintain degree plans and to enable the advising of all undergraduate computer science majors.

This significant extension will require that a database be created that will contain the essential facts obtained from each student during a consultation. The database must be updated during each consultation, and the facts must be in a suitable form for input to the expert advisor during subsequent advising sessions. Since a student who is enrolled at registration time can not be certain of his/her final grade in current classes, the database must contain a record of courses for which the student

is currently enrolled. That information will be used to query the student as to anticipated grade for each of the courses in which he/she is enrolled. Regular updating of the database must occur after final grades are recorded in order to continue to enforce prerequisites and to maintain an accurate degree plan for each student.

In order to advise students in their second (or later) semesters, it will be necessary to create a file containing course and prerequisite information for all courses taught at Lamar. Both courses and prerequisites are subject to modification each year, so a significant and continuing maintenance effort will be required as the program remains in continued used.

7. Summary

CLIPS provides a very convenient development environment. The CLIPSWIN interface is quite easy to use, and all of the documentation is clear and precise. The primary weakness, from the author's point of view, is the limited I/O capability. The user interface and report generation are awkward to construct without such capabilities as positioning the cursor and sending carriage control characters to the printer.

The author had considerable previous experience programming in LISP and Prolog, and had experimented with Personal Consultant™ Plus, but had no prior experience with CLIPS. In preparation for this project, approximately four to six hours was devoted to reading the user's guide (2) and browsing through the reference manual (3) before attempting any programming. After writing a few very short examples, mainly checking the I/O features and some special functions such as "member" and "subset", it was felt that enough had been accomplished to begin the program. Expertise in constructing complex rules was developed very quickly.

CLIPS appears to be quite suitable for use in an introductory course on expert systems in which students have limited programming experience. One or two class periods, with examples chosen from the user's guide, should be sufficient to enable the students to begin writing their own programs. More advanced students can be given the user's guide and allowed to learn in a self-paced manner.

It is intended that the expert advisor, after field testing, will be expanded to aid in the advising of all computer science majors at Lamar University.

REFERENCES

1. Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., and Sacerdoti, E.: The Architecture of Expert   Systems. In F. Hayes-Roth, D. Waterman,   and D. Lenat (eds.) Building Expert Systems,   Reading: Addison-Wesley, pp. 89-126, 1983.

2. Giarratano, J. C. CLIPS User's Guide, Lyndon B. Johnson Space Center, 1989.

3. Culbert, C. CLIPS Reference Manual, Lyndon B. Johnson Space Center, 1989.

4. Giarratano, J.C. and Riley, G. Expert Systems Principles and Programming, Boston: PWS-KENT Pub. Co., 1989.

5. Personal Consultant™ Plus (2539262-0001) Rev. C, Texas Instruments Incorporated, 1986.

6. Weiss, S. M. and C. A. Kulikowski, A Practical Guide to Designing   Expert Systems, Rowman & Allanheld, 1984.