Academic Virtual Advisor

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Kathryn Nobles

Certificate of Approval:

Gerry Dozier
Associate Professor
Computer Science and Software Engineering

Juan Gilbert, Chair
Associate Professor
Computer Science and Software Engineering

Cheryl Seals
Assistant Professor
Computer Science and Software Engineering

George T. Flowers
Interim Dean
Graduate School

# Academic Virtual Advisor

Kathryn Nobles

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
May 10, 2007

Academic Virtual Advisor

Kathryn Nobles



_____

Signature of Author

_____

Date of Graduation

# Vita

Kathryn Lynn Nobles, daughter of Gregory Allen and Terri (Nelson) Nobles, was born on December 10, 1981 in Birmingham, Alabama. She graduated from Thorsby High School as Valedictorian in 2000. She attended Auburn University in Auburn, Alabama and graduated with a Bachelor of Science degree in Computer Science in May, 2004. She entered Graduate School, Auburn University, in August of 2004.

Thesis Abstract

Academic Virtual Advisor

Kathryn Nobles

Master of Science, May 10, 2007
(B.S., Auburn University, 2004)

75 Typed Pages

Directed by Juan Gilbert

With the increasing aptitude of artificial intelligence and expert systems, new and innovative uses are being discovered. The objective of this research was to examine the use of an expert system to solve the problem of academic counseling. As the number of students on a college campus increases, the amount of time an advisor can dedicate to an individual student decreases. An intelligent interactive counseling system supported by a back-end database could decrease the amount of counseling time necessary for each student. When placed in an online environment, the system gains the ability to counsel numerous students simultaneously, in an environment comfortable to the student. This is an exploratory look at the time saved using an intelligent interactive counseling system.

## Acknowledgments

The author would like to thank her parents for their emotional and financial support during her college career. Additional thanks must also be given to the members of the HCCL lab for proofreading, advice, opinions and support. Also, thanks are due to Judy Aull and Drs. Saad Biaz and David Umphress for their support both emotionally and academically. To Dr. Johnny Green, Dr. Kai Chang, and Dr. Dean Hendrix, many thanks for allowing the author the opportunity to become financially independent for this degree. And especially to Drs. Juan Gilbert, Gerry Dozier, and Cheryl Seals for being not only good professors, but also good mentors and a tremendous help.

A special thanks is deserved by Kevin Walker, Mark Ivester, Paul Caspers, and Ches and Amy Smith for your friendship, tremendous emotional support, encouragement, understanding and chocolate ice cream. Without friends to remind you to get your work done, it might never get done.

And anyone else that ever helped the author with her homework, calmed her down, or dealt with her incessant worrying, thank you.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliograpy follows van Leunen's *A Handbook for Scholars.*

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

Table of Contents

# List of Figures

# Chapter 1

# Background

## 1.1 Academic Advising Software

In this section, previously created software to handle the task of academic advising will be discussed.

### 1.1.1 FROSH

Developed for Saint Peters College, FROSH was intended as both an advising training system and a stand-alone system for advising freshmen. It was designed using VP-Expert, an expert system development tool.

Its algorithm consisted of the following steps:

1. determine the maximum number of courses a student should take
2. choose the math and English courses for the student
3. determine the students major
4. choose an introduction course from that major OR advise the student to wait until prerequisites are fulfilled.
5. finish choosing courses until students program is complete.

Some of the shortcomings the system included not allowing for current registration information, such as conflicts in course offerings, closed sections, or any course scheduling.

When beginning the design process, the creators of FROSH outlined a set of criteria that were necessary to advising freshman. As mentioned before, it first took into account the number of courses that could be taken in the first semester. Next, they noted the courses that all freshmen take at the beginning of their academic career, then classes a student will need at the beginning of their major. And finally, the remainder of the courses needed to fulfill a course load.

As schools can sometimes offer courses with a varying number of hours per course, it was found that counting the number of courses and not the number of credit hours was insufficient. Due to this discovery an additional system (the first merely counted courses) was developed.

FROSH version 2 was developed in Visual Basic because of the popularity at the time, it's intuitive graphical interface, and it's ability to connect to Microsoft Access databases. The user inputs basic information, clicks a button, and the program presents the user with options for both general and major requirements. Another window is used to choose course sections. If the general information is not entered correctly, an error message is displayed.

After the courses are chosen, the system ensures there are no time conflicts or multiple sections of the same course in the proposed schedule.

Note that the observed shortcomings, such as the resolution of course conflicts, are present in the description of the algorithm for the program. It is unclear if these problems were resolved for the second release or were listed as merely wishful thinking [11].

### 1.1.2 A Web-Based Academic Advising System

At Florida Atlantic University (FAU) personal interactions were found to cause inconsistencies in the advising process. Most of these inconsistencies involved answering recurring questions and the poor utilization of resources among the different advisors. Therefore, they set out to research and design a system that would provide stability in advising. However, most of the web-based advising systems they found were forums, PDF or HTML official documents available for download, useful links, or some amalgamation of the three.

Through this research, several objectives were outlined for web-based advising:

- To minimize repetitive tasks currently performed by advisors
- To encourage students to adopt a proactive attitude toward advising-related issues
- To extend the availability of official advising-related information to remote students
- To provide academic guidance in a consistent way
- To make advising-related information available in a single place, in electronic format
- To maintain a (set of) HTML page(s) with the most frequently asked questions (FAQs)
- To develop a set of HTML forms and related ASP (Active Server Pages) scripts that allow a student to input the courses they have taken, press a button ("Advise Me") and get a list of courses to take next

The resulting program was created using HTML, forms and ASP scripts. From it's main page a user can access the requirements for their degree, a career guide, information pertaining to advising, and frequently asked questions. Most importantly, the user can access the form which allows him/her to input course information and personalized advice.

The system supports three types of users: student users, faculty users, and administrative users, each with a different graphical user interface (GUI), appropriate rights and privileges, and set of actions. The students will use the system for advice, the faculty will update and manage information relevant to the FAQ page, and the administrative users will be responsible for the next courses to take module. All information that is regarded as classified or sensitive is password protected.

Within the system there are 2 modules: FAQ and the next courses to take (hereafter known as 'Courses'). The FAQ module is a dynamically generated page that uses a backend database maintained by advisors. Questions are sorted across three categories: general, CS-specific, and CE-specific. Each question is input into the database with a unique key (identifier), category, question, answer, a date representing when it was last updated, and the name of who did the updating.

The Courses module is designed to resemble the hard copy worksheets that are preexisting within the FAU CSE department. There are also three types of worksheets correlating to four-year students, transfer students, and second bachelor students. The backend database for the Courses subsystem consists of two tables, CourseInfo and Prerequisite. CourseInfo contains the course number, prefix, description, number of credit hours and type, which are the same three types previously mentioned for FAQ questions. Prerequisite has two input fields: one for the course to be taken and one for the course that is its prerequisite.

After the Course subsystem retrieves input on courses available to be taken, it builds a directed graph based on the prerequisite information and does a topological sort.

The designers of this system found several benefits to it. Not only did it increase the ability to access official information, but it also allowed answers to be found in a timely

manner to most questions. Additionally, it decreased the amount of time advisors typically spent on recurring tasks along with a reduction in inconsistent advising [8].

### 1.1.3 IUA

This software was developed for use at Sam Houston State University (SHSU) to combat the increasing workload expectation upon faculty and staff. Since SHSU is largely a teaching university, professors are often called to not only teach twelve hours per semester and conduct research, but also to advise students on their progress towards graduation. As the number of students increases, this becomes a more difficult situation.

Students at SHSU are also allowed to enroll without an advisement period, which also leads to problems retaining students.

Initially implemented within the Computer Science Department at SHSU, it was actually expected to be used across the university once beta testing was completed.

SHSU found that students do not like to be required to be advised in order to register. Particularly, students that must commute to school do not like a policy involving mandatory advisement. However, if not required to, many students will not seek advisement and, thus, delay their graduation due to poor choices. Often a student in this situation will reach the upperclassman level without completing the necessary prerequisites that should have been taken during their freshman year. It was decided that students, regardless of maturity level, need guidance when it comes to a course of study. It is noted that "nontraditional students", defined as students over 25, also have problems reaching a degree due to faulty academic advising.

Therefore, an Intelligent University Advisor (IUA) was designed to allow students access to academic advising anywhere on the department's LAN. Designed as a tool to assist, not replace, IUA advises students with more complicated problems to continue to seek a faculty member for advice. The IUA will provide additional options to the aforementioned commuter student who would otherwise have to travel to campus for crucial academic support. The IUA, if used as designed, will allow much more freedom for both students and professors because it will not restrict academic advising to the free time of a students preferred professor.

IUA uses a Microsoft Access database to maintain information on colleges, departments, degrees, courses, prerequisites, and other pertinent information. Each department has a single individual who is given write privileges to information related to that department. Each department representative is responsible for initial inputs to the databases as well as maintaining said information via a generic GUI. By having all information stored in a database instead of hard-coded, it allows the system more flexibility and the departments remain in control of their respective data.

Additionally, the system maintains the system date every time a plan of study is altered. Because students are subject to requirements of a degree based on the time they enter school, not when they leave, this allows the system to give higher quality advice that is more tailored to each individual student.

IUA supports Texas's system of transferable transcripts and does not require any user input regarding courses they have completed. There is a feature which allows the user to enter that information if he or she is unable to locate their transcript otherwise (for instance, the case of an out of state transfer to SHSU).

IUA, through its GUI, begins with the student inputting their ID. If they have an existing transcript, it is loaded. If not, a list of possible majors is presented to them along with other information such as department, type of degree, and minor. IUA then creates the page necessary for advisement with all necessary information included. The student is then able to print the sheet.

There is an additional prerequisite sheet available to the student, which details all of the courses the student has yet to complete the prerequisites for and what those prerequisites are. The intentions of this sheet are to avoid an ongoing problem of students enrolling in courses for which they were not able to take.

In the case where a simple traversal of courses is not enough, an advisor must alter the requirements for the student via their 'write privileges'. These changes are automatically saved for the student such that the advisor does not have to repeat this action every semester. A similar course of events involves the transfer students. The main difference being that the system learns from the changes placed upon it and makes relevant suggestions in the future.

This aspect of learning is crucial to the 'equivalences' in degree requirements among various universities. These equivalences are in constant flux and thus typically require a tremendous amount of paperwork for the advisors. Additionally, if a department, like SHSU, is advising it's students as a faculty and not with a dedicated advisor, it is difficult to express the changes to every member of said faculty. Thus, if the department advisor of IUA makes changes on an 'as needed' basis and approves of all suggested equivalences. then a more uniform decision can be assured across departments and universities.

In the discussion of IUA's antagonists, it is noted that most students are often plagued with intimidation of advisors and faculty and thus less likely to seek their advice. This is presented to negate the concept that only students who want to be advised will accept an advising program. It is also touched upon that the academic advising software's non-personal nature can work just as much for it as against it, as some students desire a business like relationship with an advisor rather than a personal one that often exists [3] .

### 1.1.4 ADVISER

Designed at the University of Wisconsin in 1968, ADVISER was one of the first programs of its kind. ADVISER was programmed in ALGOL on 3000 cards with 22 methods. It was developed not only to deal with the University of Wisconsin's course requirements, but to also handle the equivalencies generated by transfer credits. ADVISER is also not only for undergraduates, but also advises graduate level students, an aspect not often duplicated in other advising software.

According to its algorithm, ADVISER first conducts an interview with the student to gather information on the student's completed courses and to conduct educated guesses concerning courses the student is unsure of having taken. It then calculates what the student's course load should be. However, it uses a great deal more math and statistics to determine a suggested course load for each student than other similar software.

Adviser did have a study conducted on it. The study had eleven participants from various degree programs within the computer science department at the University of Wisconsin. Although the study had a small number of participants, the testers felt the diversity

of the pool made up for it and validated their results. The study concluded that the interview process was far too long, that most participants were satisfied with the program, and that all would use it again if it were kept up-to-date. However, it was also found that not all enjoyed using it and that most of the subjects that were in graduate school did not enjoy using it and some were dissatisfied with the system.

Another important conclusion was the most students preferred a human advisor to a programmed one. Since advice is a subjective matter and cannot be measured holistically concerning its quality, it is impossible to determine if the advice of the computer was better or worse than the advice of a human advisor. Therefore, the only measure of quality is that of the student's perception, which is clearly slanted towards the human advisor, based on these results [12].

### 1.1.5 DSS for Academic Advising

This Decision Support System(DSS) was implemented to allow human advisers to focus on the more complicated problems rather than the more algorithmic course load selections. It takes into account the four types of academic courses: university requirements taken by all students (courses such as English, mathematics, and history), core requirements taken by students within a wide area (such as a college), major requirements, and electives. Unlike most of programs of its nature, this advising software was designed initially for business students rather than engineering, more specifically computer science.

A DSS is presented as an easier way to evaluate a student's progress towards graduation. It can also be a quick way to not just list courses that are required, but also those that the

student is allowed to take in that the student has completed the necessary prerequisites. This calculation must be an error-free one for the system to have merit.

Since academic advising is typically a very structured process with the selection and sequencing of courses, the concept of a DSS can easily be applied. Additionally, an expert system (ES) can be used because the problem scope is quite narrow. Both methods include similar components of a knowledge base, an inference engine, and a user interface.

Typically, an advising support system will use the plan of study for a major, taking into account the optimal scheduling to minimize semesters in school. This ignores course content and individual student issues.

This particular Academic Advising software requires the student to input their own course information each time the system is used. Beyond this step, the program is designed quite similar to other advising systems. After the student has input his/her completed courses, the DSS produces a list of eligible courses and completed courses using binary categories within the database.

Also used to choose eligible courses and more specifically, their order, are three hierarchical rules. The first rule is the 'Deepest Layer Rule' which chooses courses on the deepest level of prerequisites first and then choose courses based on the descending order of their layers. Next is the 'Maximum Dependency Rule', which sorts courses within each layer by the number of prerequisites they will complete. Lastly, the 'Course Number Rule', which chooses courses based on the ascending order by their course number.

It is concluded that this DSS will work, but is not the best option. Instead, a database management system is recommended. Since the database is the largest part of the DSS it becomes difficult to separate it from the inference engine, a DSS generator would create

a more flexible user experience as well as provide an easier method of maintaining degree requirements [9].

## 1.2 Expert Systems

The following sections give a brief description of some existing expert systems.

### 1.2.1 Knowledge Engineering and Expert Systems

An expert system can be thought of as a program with two components: a ruleset (RS) and an inference engine. The RS consists of the information that the inference engine will process. Each piece of the RS typically contains two parts: the antecedent (ant) or condition and a conclusion (cul) coupled with a probability (prb). A rule then looks like Figure 1.1.

$$ru_i=[(ant_1 \text{ v } ant_2 \text{ v } ,,, \text{ v } ant_n)$$
$$(cul,prb)_1\ (cul,prb)_2\ ,,,\ (cul,prb)_m]\ (1\text{-}1)$$

Figure 1.1: Example of a rule in an expert system[4]

Because the rules form a set, these rules must be "syntactically different". The antecedents must be both sensitive and selective to insure that a conclusion will be "triggered" and that it is the correct conclusion.

The inference engine is comprised of two pieces, a pattern-matcher and a conflict-resolution procedure. A basic approach would involve pattern-matching the antecedents of the rule with any new information found. If a pattern is found, the antecedents "triggered rule is added as new information. If it finds two rules triggered simultaneously, it uses a

priori criteria to obtain a conflict resolution and get the best choice. After these steps, the process repeats with the updated data.

This system is meant to emulate human cognitive abilities. Because the results of a triggered rule can then become information later used to trigger another rule, the system produces a casual relationship to the data. One could even argue that it "learns" how to better deal with certain behaviors, similar to how a human learns which foods they like and which ones make them sick and how later on, they use this knowledge to avoid certain foods.

Knowledge engineering refers to the process of creating an RS. Between acquiring the knowledge, testing it, and evaluating it, the knowledge engineer determines the rules and makes sure these new rules to not produce unanticipated problems with pre-existing rules. This is continued until there are no new rules [4].

### 1.2.2 Expert Systems and Intervention Styles

Focused on analyzing the benefits of a decision support system (DSS) integrated with expert systems and applied to intervention consulting, the authors focused on the evolution necessary for this facilitation. Different styles of interaction (directiveness and nondirectiveness) are also highlighted.

A DSS is thought of as a way to use a standard practice on categorized information for which the user must determine the course of action and the quality of its results. An expert system often is a more lithe solution but cannot be used for a partial specification. The definition of both systems of problem resolution led to their reapplication in order to form a more flexible solution. The expert support system (ESS) maintains the expert systems

flexibility and incorporates the DSSs user specifications. Thus, it is the ESS that is chosen to work with intervention styles.

Several factors are laid out to easier understand intervention styles and the choices described. These factors include the "cognitive state of the user" (defined as learner, solver, or refresher), restrictiveness of the software environment, a users posture, and the time pressure placed on the user.

Restrictiveness of the software was found to hinder guidance in more restrictive systems. Time pressure is shown to be directly related to the directiveness and dominance.

Several intervention styles are described. For the directive method of intervention, the consultant tends to lead the conversation, whereas in the nondirective form, questions are asked to facilitate the clients participation in the problems ultimate solution. Within the directive approach, there exist variants: acceptant, catalytic, confrontation, prescription, and theory-based.

Acceptant is more people driven, and the least directive of the five directive styles. Control is given to the user and the user maintains the ability to choose his/her own process and content. Acceptant is considered ideal for learning with openness being an important factor. Eventually the user will become extremely satisfied and no longer want the system to resolve the issues that drove the user to it. Open ended questions and electronic brainstorming tend to fall within this style.

Catalytic directiveness is a compromise between people and results. Using this approach, the system aids the user to accumulate information and then infers a different opinion of the problem. However, this style rarely increases the user's understanding of problem solving.

Confrontational intervention examines the users assumptions. This style has no alternative to offer in regards to its possible failings, a major problem.

Prescriptive styles tend to cause a user dependency. Because the system merely provides solutions to problems without encouraging the user to think about the problem itself is a significant problem. This intervention style is best applied to users with high time-pressure and are using a highly restrictive system. Such intervention techniques are often applied to things such as intelligent knowledge-based tutoring systems where the content and process are highly controlled.

Theory-based intervention is the most analytic, teaching the user to evaluate the problem and prepare better for it [15].

## 1.3 Case Based Reasoning

This section describes several different approaches to case-based reasoning.

### 1.3.1 CBR and Deep Structure Approach to Knowledge Representation

This system uses case-based reasoning (CBR) to determine actions relating to legal situations. The theory behind this plan is that CBR is a closer match to the reasoning of an actual lawyer. Also, as new cases are entered, the decisions will evolve, much like the nature of law.

From its beginning as a predominantly rule-based system, it, too, evolved into a CBR system using its rules as a new system of knowledge. After many of these initial rules were removed, CBR rules to control case retrieval were inserted. However, this does not affect

its expertise. It is possible that the 'ghosts' of each former rule still held some influence on new cases due to their influences on previous cases.

The database used in this system contained 'profiles' of each case. The profiles consisted of all facts needed to decide a case and a slot for a record of its resolution.

Based on the user's input of a case, it then uses the 'horizontal search method' to match the user's case to previously resolved cases. In this method, facts are compared between the cases and the one resolved case with the most similar facts to the user's case is considered a match.

In its evaluations, it is stated that a purely rule based system would not allow for the easy manipulation of knowledge, something very necessary in a legal system. However, a CBR approach allowed for a greater freedom concerning knowledge representation and alteration [7].

### 1.3.2 Monological Reason-Based Logic

Also designed to assist with legal cases, this system uses an amalgamation of CBR and rule based reasoning (RBR). It uses the rules of the RBR and determines the scope of those rules using CBR.

Reason based logic (RBL) acquires the reasons, which consist of facts, that argue for and against a thesis and then uses these to evaluate the thesis based on the reasons.

The author states that his personal belief is that CBR is about weighing reasons. This is explained by showing how a lawyer would argue that his/her point should be valid because it was valid in a previous case.

Another aspect of reasoning systems is that they may not always reach a conclusion. If there are equal amounts of reasons on both sides of an issue, the algorithms will not find either answer better than the other. Additionally, if the reasons are in conflict with themselves, an answer will not be found [5].

## 1.4 Academic Advising of Computer Science Students

### 1.4.1 New Approaches To Advising and Mentoring in Science and Technology

At Texas A and M University (TAMU) students once had a familiar sounding advising and registration process. A hold was placed on each freshman until they saw an advisor and their schedule was chosen and registered.

However, it was found that a new model needed to be established. This new model did not simply choose a schedule for them, but taught them how to create one later. It was also noted that students benefited greatly from the advice of other students currently enrolled in the questioned courses. Students were also encouraged to meet with faculty members after the initial design of their schedules [1].

### 1.4.2 Academic Advising is Not Rocket Science

After observing the process of academic advising for over twenty years, Woolston theorized that the problem with advising was not an information organization problem, but rather one of interpersonal relationships. "Engineering is a kingdom of facts, not opinions he stated.

Presented are several cases where Engineering advisors perceived themselves as doing an excellent job through flow charts and comprehensive websites, yet students still were

critical of the process. Therefore, he hypothesized that what students really want are opinions and not the "magical combination of courses to complete their degree [14].

### 1.4.3 Evolution of Academic Advising

The Department of Electrical and Computer Engineering at Kansas State University wanted to adjust its advising process in 1992. They wanted to account for the students that needed extensive advising and also the students who needed a prerequisite check.

They formed a committee to redefine their process. That committee developed flow charts of prerequisites, letters and postcards to inform students of the process, sign up sheets, and student counseling forms. Also, an academic progress committee was formed to assist students who were not making satisfactory progress towards graduation. This new committee included more than one faculty member who sat down with a student for a short, more formal discussion of their academic situation.

It was found that students were pleased with a flexible advising process that allowed them to spend as little or as much time as they wanted on the process. Faculty also approved as it greatly decreased their time commitment. It was also stated that a database was planned to further assist the prerequisite checks, allowing automation of that process instead of the existing system of underlining and circling [6].

# Chapter 2

# The Program

## 2.1 A Problem and a Solution

In a perfect world, there would be one advisor for every student at every collegiate campus all across the globe. One advisor to ensure that each student not only made satisfactory progress towards graduation, but tailor made the students academic schedules to best suit the student.

But we do not live in a perfect world. We live in a world where the students vastly outnumber the academic advisors. With such a disproportionate number, time is of the essence. Advisors must find a way to determine each students perfect schedule for typically hundreds of students.

Additionally, academic advising for an entire university often occurs in less than a month. Therefore, most students must find time in the approximately twenty business days to meet with their advisor, often being forced to meet with them before they are able to register for classes.

This limits each student to five to ten minutes to determine the next six to eight months of their academic career. And yet it often takes a student an entire afternoon to line up a possible schedule.

It is not uncommon for student records to be kept in a different building than the building within which a student will be advised in. Furthermore, it is often the student who must retrieve their own records and present them to their advisor. As the student does not hold a key to their own records, they must wait in line to have them located, trek across

campus to their advisor where it is typical to wait in line again. This can become quite frustrating.

But as technology becomes more prevalent and campuses become more wired, the registration lines of not too long ago seem archaic. And yet, if it is possible to register online and course information is already stored in a secure database, why is it that students still must wait in line to be advised?

The Academic Virtual Advisor (AVA) was designed with this in mind. Intended as a tool to alleviate the overcrowding of advisor offices, AVA can be used to supplement existing advisors. By using existing databases that contain student information and allowing advisors to create new databases stipulating available courses, course prerequisites and plan of studies for their departments (potentially in a less hectic portion of their schedule), AVA can be a surrogate advisor to most students.

AVA was also designed to leave the academic advisor in control of the advising process. While it can be used alone to assist a student, the advisor may choose to approve each potential schedule before a student registers. In the case of automatic approval, although not intended, AVA could be used to serve as a temporary advisor if an institution is currently lacking in human advisors.

However, it is the web-based aspect of AVA that will assist human advisors and students the most. Since AVA is an online, database driven system, it is capable of supporting more than one student advising session at a time. Furthermore, each student has the ability to be advised where it is most convenient for them. By using the aforementioned existing student information databases, AVA eliminates the wait time students incur to retrieve their records and to be advised. Also, it eliminates the transit between record offices and advisor offices.

Using AVA, a student can be advised in as little or as much time as they would like, but is not forced to cancel their plans for an entire afternoon.

In short, AVA is a customizable solution to the ever-increasing advisor to student ratio.

# Chapter 3

# Implementation

## 3.1 User Interface

### 3.1.1 Student Login

Before the decision was made to use the database to mock existing student record databases, a student had to input their chosen major and the semester for which they would like to be advised. With the additional use of the database, a student can now login as if they were to log in to Auburn University's OASIS system. This is shown in Figure 3.1.

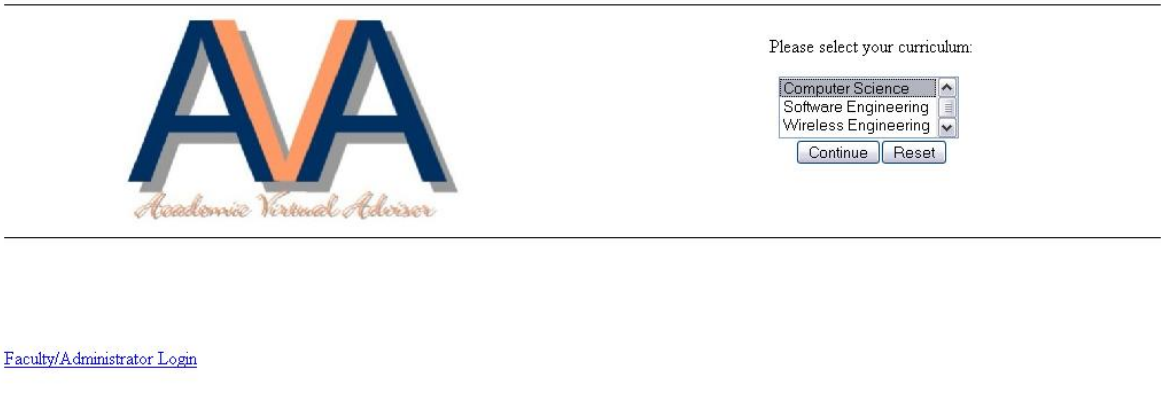



Figure 3.1: AVA Original Login Page

Figure 3.2 shows the changes made to allow an easier full adaptation of the system in the future. A system that takes advantage of input patterns students are already familiar with will cause the student to spend less time learning the system.


Figure 3.2: Current AVA Login Page

### 3.1.2 Display Course Information

Initial plans for the course information interface were to closely mimic the forms given to students and advisors in order to approve their schedule. This form involved dividing the courses into a plan of study designating which courses were to be taken during which semester. This page was originally laid out in simple HTML and the results are shown in Figure 3.3.

However, with the addition of the PHP code needed to retrieve all the information from a database, the "simple" HTML became more complex. An additional attribute in the database to keep track of which column proved necessary to achieve the look of the original page. Yet, even with the new attribute, the page never maintained the look of the original HTML page or the plan of study form. Additionally, this new attribute could be

**Bachelor of Science in Computer Science Curriculum Checklist**

Please check all courses you have completed or will have completed before the purposed schedule. If you have transfer credit, please check the equivalent of that credit.

**Freshman Year**

| Fall Semester | | | Spring Semester | |
|---|---|---|---|---|
| ENGL 1110 English Composition I | 3 | | ENGL 1120 English Composition II | 3 |
| HIST Core History (pops up box if checked) | 3 | | HIST Core History (pops up box if checked) | 3 |
| MATH 1610 Calculus I | 4 | | MATH 1620 Calculus II | 4 |
| Science Sequence I* (pops up drop down box of approved courses if checked) | 4 | | Science Sequence II* (pops up drop down box of approved courses if checked) | 4 |
| ENGR 1110 Introduction to Engineering | 2 | | **(M)COMP 1210 Fundamentals of Computing I** | 3 |

**Sophomore Year**

| Fall Semester | | | Spring Semester | |
|---|---|---|---|---|
| ENGL 2200 World Literature I | 3 | | ENGL 2210 World Literature II | 3 |
| Core Social Science Group 1(pops up box if checked) | 3 | | Core Social Science Group 2(pops up box if checked) | 3 |
| COMM 1000 or ROTC (pop up box) | 3 | | ELEC 2200 Digital Logic Circuits | 3 |
| Concentration (pops up text box if checked) | 3 | | MATH 2660 Linear Algebra | 3 |
| **(M)COMP 2210 Fundamentals of Computing II** | 3 | | **(M) COMP 2710 Software Construction** | 3 |

**Junior Year**

| Fall Semester | | | Spring Semester | |
|---|---|---|---|---|
| STAT 3600 Probability and Statistics | 3 | | PHIL 1040 Business Ethics | 3 |

Figure 3.3: Original AVA Student Information Confirmation Page

confusing to a potential administrator of the system as it would simply be used for design. Therefore, the initial user interface was put aside and a new plan was formulated.

The second plan to display course information involved three columns, "previously taken", "available to take", and "required, but not yet available to take". Previously taken courses were displayed with a marked checkbox to their left and are printed in black. Available Courses were displayed with an unmarked checkbox and were printed in blue. Unavailable courses were printed in grey. This is shown in Figure 3.4. The use of different colored text would allow the user to further distinguish between the three columns.

After a design review, the confirmation page was again changed as shown in Figure 3.5. This iteration presents the user with a more simplistic view of the information. The user can click on a link that corresponds to the course information they would like to review.

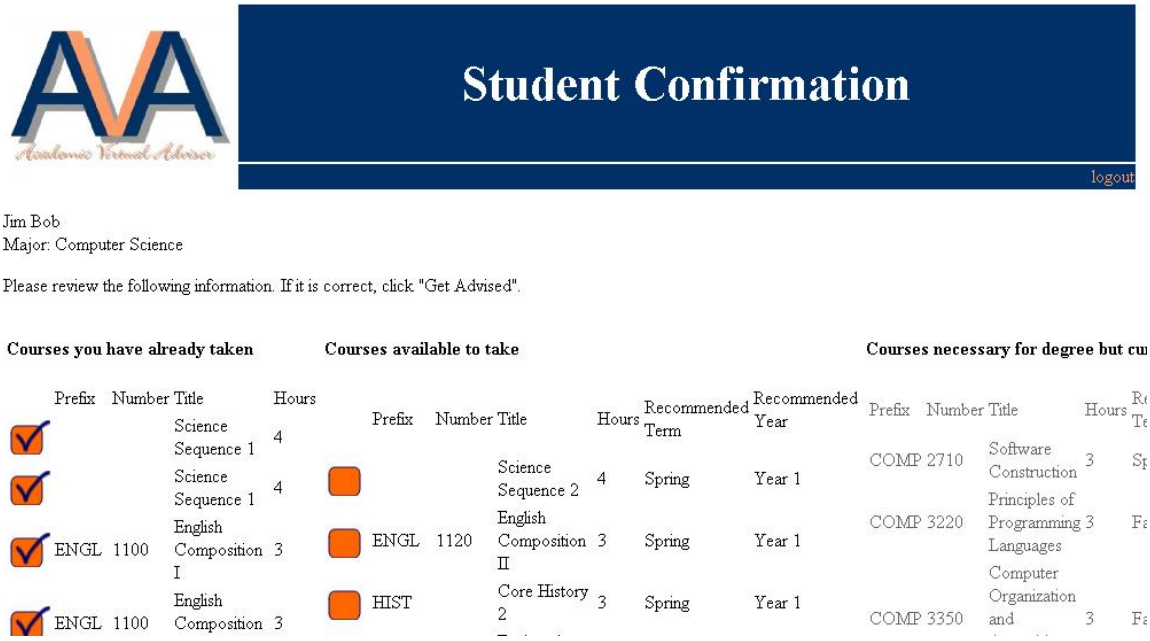



Figure 3.4: 2nd Iteration of AVA Student Information Confirmation Page

Each link directs the browser to open a new, smaller window with just that information as shown in Figure 3.6. Also, the links are color coded. The link for courses taken is shown in black. The link for available courses is shown in green, to mimic the "go" aspect of a traffic light while the link for unavailable courses is shown in red. This method also allows the user the ability to access the information as they review their proposed schedules if they choose.

### 3.1.3 Advised Schedule

The proposed schedule page was designed to look similar to the existing scheduling documents found in the Computer Science and Software Engineering Department at Auburn University. Every other line of the proposed schedules has a gray background, as shown in Figure 3.7.

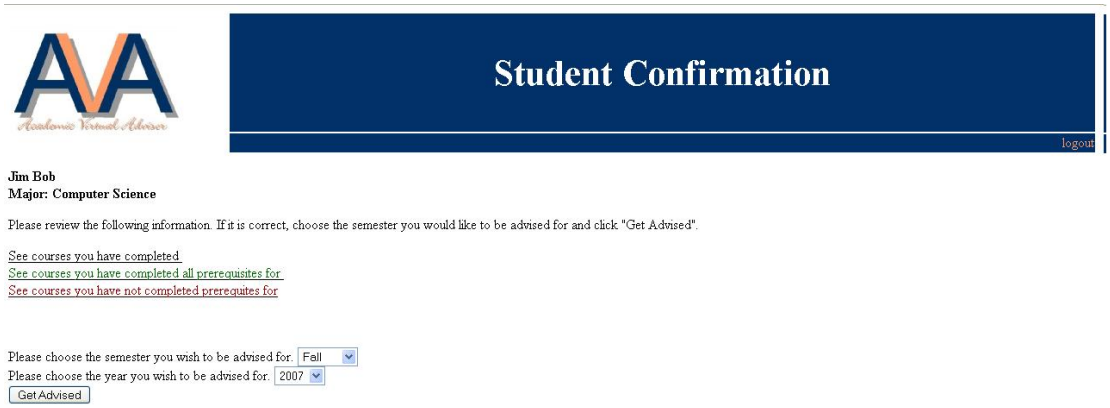



Figure 3.5: Current AVA Student Information Confirmation Page

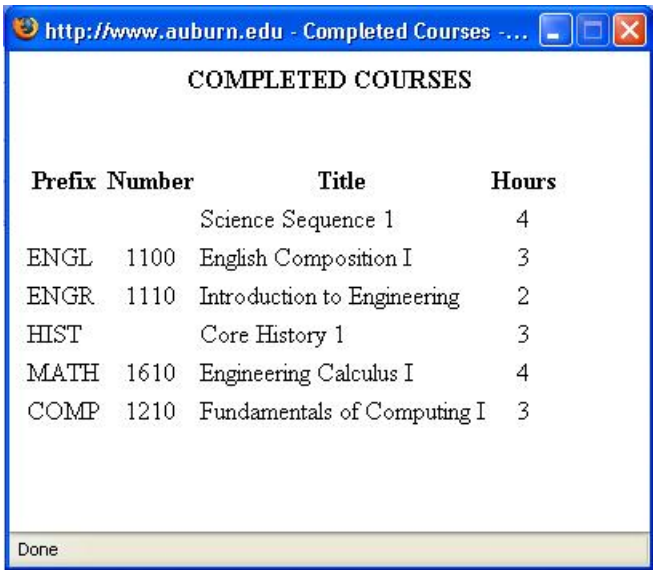



Figure 3.6: AVA Student Courses Page

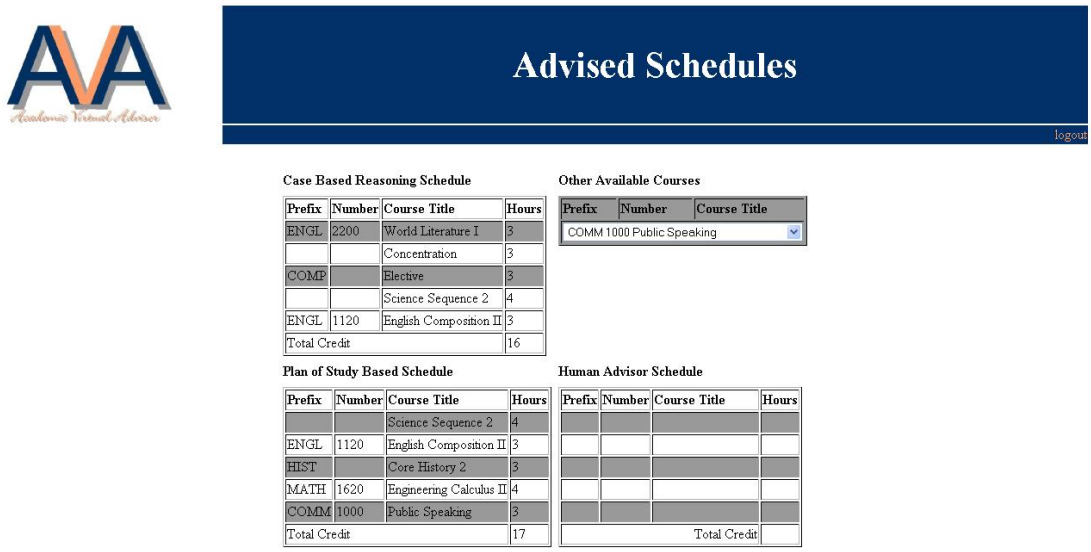



Figure 3.7: AVA Student Scheduling Page

Two schedules are displayed: one using case based reasoning and one that maps the student's taken courses to the plan of study and retrieves the next five courses in the list. Each schedule contains five courses based on the principle that most courses are given three credit hours and a "full" course load is considered to be fifteen hours.

To the right of the case based schedule is a drop down list of all remaining courses (based upon the plan of study schedule) for which that the student qualifies. This is to assist with a possible third schedule of the student's creation. To the right of the plan of study based schedule is an open schedule for either a human advisor or the student to create the third option.

### 3.2 Backend Database

In order to implement an advising program, it was actually necessary to maintain a dual purpose database. The database must serve as a system to store student information, a purpose filled at Auburn by the OASIS system, and as a system to record information about major requirements. The student information portion was filled with dummy data whereas the course information database uses the courses and requirements to earn a degree in Computer Science from Auburn University. The ER diagram shown in Figure 3.8 displays the database's relationships. The schema for the database, Figure 3.9, expresses the attributes used for each table.

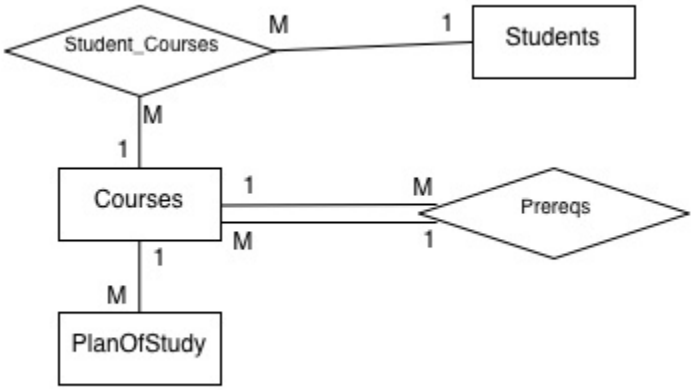



Figure 3.8: Backend Database ER Diagram

The ER diagram presents an interesting relationship, that of the Courses Table and the PreReqs table. When you look at the schema for the database, you see that the only data stored in Prereqs is the ID of the course and the ID of its prerequisite. One course may have many prerequisites and one prerequisite may be needed to take many courses. With this logic, Prereqs is actually the relationship between the Courses table and a second

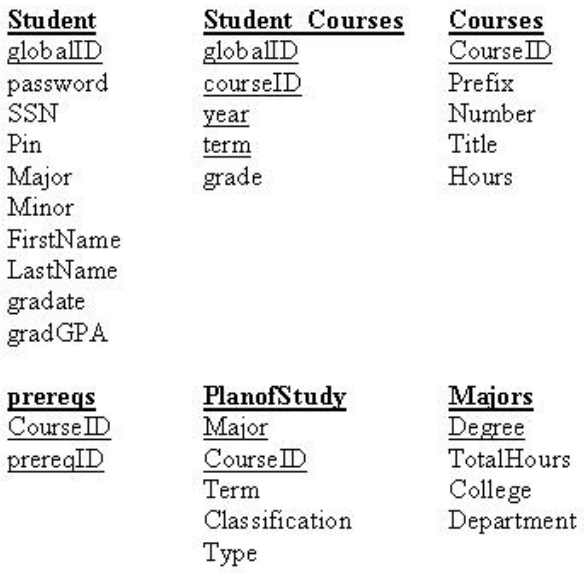



Figure 3.9: Backend Database Schema

instance of the Courses table, as a prerequisite is defined as a course which must be taken before other courses can be taken.

The backend database combined with PHP actually does most of the work for the program. Once the student's identity is verified with a query to the database, the system executes three queries to determine the nature of that student's plan of study. Figure 3.10 shows the query necessary to locate which courses the student has already taken. Figures 3.11 and 3.12 display the courses that have not been taken. Figure 3.11's query handles the courses that the student is able to sign up for while Figure 3.12's query yields what courses the student has yet to qualify for, with respect to prerequisites. Both of the queries relating to untaken courses contain three subqueries, including the subquery to determine what the student has already taken in both cases.

```
SELECT C.prefix, C.number, C.title, C.hours, p.classification, p.term
FROM student_courses S_C, PlanOfStudy p, students S, Courses C
WHERE C.courseID = S_C.courseID
AND C.courseID = p.courseID
AND S.globalID = S_C.globalID
AND S.SSN = '".$SSN."'
ORDER BY p.classification, p.term, C.prefix, C.number;
```

Figure 3.10: Completed Courses Query

```
SELECT DISTINCT c.prefix, c.number, c.title, c.hours, p.classification, p.term
FROM student_courses s_c, PlanOfStudy p, students s, Courses c
WHERE c.courseID = p.courseID
AND s.major = p.major
AND s.ssn = '".$SSN."'
AND (
c.courseID
IN (

SELECT DISTINCT pr.courseID
FROM student_courses sc, students s, prereqs pr
WHERE sc.globalID = s.globalID
AND ssn = '".$SSN."'
AND pr.prereqID = sc.courseID
)
OR c.courseID NOT
IN (

SELECT DISTINCT courseID
FROM prereqs
)
)
AND c.courseID NOT
IN (

SELECT DISTINCT sc.courseID
FROM student_courses sc, students s
WHERE sc.globalID = s.globalID
AND ssn = '".$SSN."'
)
ORDER BY p.classification, p.term, c.prefix, c.number
```

Figure 3.11: Available Courses Query

```
SELECT DISTINCT c.prefix, c.number, c.title, c.hours, p.classification, p.term
FROM student_courses s_c, PlanOfStudy p, students s, Courses c
WHERE c.courseID = p.courseID
AND s.major = p.major
AND s.ssn = '".$SSN."'

AND
c.courseID
IN (

SELECT DISTINCT courseID
FROM prereqs
)

AND c.courseID NOT
IN (

SELECT DISTINCT pr.courseID
FROM student_courses sc, students s, prereqs pr
WHERE sc.globalID = s.globalID
AND ssn = '".$SSN."'
AND pr.prereqID = sc.courseID
)

AND c.courseID NOT
IN (

SELECT DISTINCT sc.courseID
FROM student_courses sc, students s
WHERE sc.globalID = s.globalID
AND ssn = '".$SSN."'
)
ORDER BY p.classification, p.term, c.prefix, c.number
```

Figure 3.12: Unavailable Courses Query

Once the student proceeds to the next step, only the query yielding the currently untaken, but available courses will be needed. It, along with the query in Figure 3.13, is used by the computer to determine which courses should be taken, forming the two possible schedules. The unused results from this query are listed in order to browse for the student to make their own schedule.

## 3.3 Schedule Creation

### 3.3.1 Case Based Reasoning

AVA uses Case Based Reasoning (CBR) to create one of the proposed schedules. As AVA was built with a database designed to simulate a real database of student records, fictitious data was created and inserted to provide cases. These fake records are full records of what classes the student took, when, and what grades they earned in that course. It also stores when they graduated and their GPA upon graduation, if appropriate.

After the student logs in, the system retrieves their completed courses, the grades they received, and when they completed the courses. It then matches, based on time and grade earned, those courses to another full record of a graduated student. Using the most accurate match, the student is then advised to take the courses that the matched case took to complete the curriculum.

These recommendations are checked to make sure the student has not already taken them. If so, they are no longer included in the possible schedule. This may leave the proposed schedule with fewer courses than needed to maintain a full load. In this case, the schedule is completed using the next courses in the chain made up by the plan of study.

The query for the CBR schedule is shown in Figure 3.13.

```
SELECT sc2.globalID, count( sc2.courseID )
FROM student_courses sc1, student_courses sc2, students s, students s2
WHERE s.ssn = '".$SSN."'
AND s.globalID = sc1.globalID
AND sc1.courseID = sc2.courseID
AND sc1.year = sc2.year
AND sc1.term = sc2.term
AND sc1.grade = sc2.grade
AND sc2.globalID = s2.globalID
AND s2.gradDate IS NOT NULL
GROUP BY sc2.globalID
ORDER BY 2;
```

Figure 3.13: Case Based Query

### 3.3.2 Plan of Study Schedule

A plan of study can be thought of as a roadmap to graduation. In this situation, the students completed courses are considered nodes along that path. The completion of a node opens up other nodes as possible options (by completing prerequisites).

In the plan of study based schedule, the student's completed courses are compared to a listing of prerequisites. The courses that have their prerequisites met are listed in order of semester and year the plan of study advises them to be taken minus the courses that have already been taken.

The schedule is then formed by choosing the first five courses from this list. The query for the plan of study based schedule is shown in Figure 3.11, it is the same query used to determine available courses during the information confirmation step.

# Chapter 4

# Usability Study

## 4.1 Materials

A study was conducted at Auburn University in the Human Centered Computing Lab. Participants sat at the same desk and used the same computer, an IBM CPU with Microsoft Windows XP. The computer operated with a Dell monitor, IBM mouse, and an IBM standard keyboard. The system was run using Mozilla Firefox version 1.3.0.10.

## 4.2 Participants and Procedure

Twenty-Five college level students participated in the study. All participants had the following controls applied:

1. they sat in the same chair
2. they completed the same tasks in order
3. they sat in the same room along with the researcher
4. all participants were told not to discuss the study with anyone thus eliminating a disparity in knowledge of the experiment.

A total of ten fake students were created within the database. Three of these were given complete records including a graduation date and a GPA at graduation and the other seven were created with partial records to use within the usability study. Each of the

participants was instructed to choose one of these six fake records to use as if they were their own records.

Each participant then logged into the system as described above, was shown the previously mentioned fake records to review as they saw fit, and then proceeded to be advised for the next semester's schedule.

After each subject finished being advised by the system, they were given a questionnaire, found in *APPENDIX D*, containing thirty-two questions for which to rate its usefulness and appeal.

### 4.3 Data Collection Method

The questionnaire, given as a post-questionnaire, gathered information about the participant and about their experience with AVA. It contained thirty-two questions including multiple choice questions (used to gather information about the user) and bi-polar rating scales (used to ascertain the user's satisfaction with the system). Details of the questionnaire can be found in *APPENDIX D*.

### 4.4 Results and Discussion

This section describes the quantitative data gathered from the questionnaires concerning participant background and their responses to the system.

### 4.4.1 Participant background

As shown in Figure 4.1, the study contained 8.33% females and 91.67% males. 95.83% stated that English was their native language and 4.17% stated that it was not. All participants were enrolled in ENGR 1110: Introduction to Software Engineering in the Computer Science and Software Engineering Department at Auburn University; therefore, a general comfort with computers could be inferred.

| | |
|---|---|
| Male | 91.7 % |
| Female | 8.33 % |
| English - 2nd Language | 4.17 % |

Figure 4.1: Participant Background

### 4.4.2 User Satisfaction

The questionnaire allowed users to rate their experience using a 10-point bi-polar scale. For most questions, a higher value dictates a more positive experience and a lower score implies a more negative experience. However, one question did not exhibit this pattern: *ease of operation depends on your level of experience*. For this question, the inverse of the typical pattern is exhibited in that a higher value presents a more negative experience. The charts containing frequencies for all questions concerning user satisfaction found on the survey can be located in *APPENDIX A*. The tables with displaying the mean, median, maximum, minimum, and standard deviation of each question's responses can be found in *APPENDIX B*.

The most important aspect to investigate was whether or not students would be willing to use a system such as AVA in the future. Figure A.28 shows that over 70% ranked this

with a 7 or higher. Those that ranked its possible future use with a 9 also ranked its straightforwardness, speed, and ease of task completion with an 8 or higher.

It is possible to conclude that this is reasonable as the students most likely to use a virtual advisor might be disenfranchised with the current process's time consuming nature.

Participants that ranked possible reuse with an 8 ranked understandable prompts and easy of getting started with an 8 or higher.

The next aspect to investigate was the user satisfaction with the two prepared schedules. 75% of participants ranked the approval of the prepared schedules with a 7 or higher, shown in Figure A.29. However, nothing else can be concluded by casual observation.

An interesting observation is that participants that ranked the system overall wonderfulness with an 8 or higher, also ranked their overall satisfaction with the system with an 8 or higher. This could mean that this subset of participants equates wonderful with satisfactory.

The final question on the questionnaire was open-ended such that participants can remark on the things they felt were important, but weren't covered by the questionnaire. Of the twenty-four participants, fourteen took advantage of this opportunity. One participant stated that he/she would "prefer experienced professional flexible counselors to rigid programmed algorithms." This response further supports the theory that while most students might find an online advising system "extremely helpful", as two participants stated, not all students would.

Other comments included "easy to use", stated by three participants, "well organized", again by three participants, "to the point", by two, "easy to understand", and "would recommend using". Also, one student, with no knowledge of the reasoning behind the

program's login features, as stated in Section 3.1, responded that he/she would also like to use it to enroll. This possibility will be further discussed in Section 7.

Another interesting set of comments was related to the "links" used to call the JavaScript to display course information confirmation. Three participants remarked that it was difficult for them to understand that they could click on them because the cursor did not change, indicating a link. One student even went so far as to suggest that it appeared more like a database function than a link.

# Chapter 5

## Schedule Comparison

For the purpose of this section, the phrase "CBR failed" is to imply that CBR contributed nothing to the CBR based schedule and needed the plan of study based schedule to fill in gaps.

While impossible to tell if one schedule is truly better than the other, it is possible to determine their similarities, faults relating to each other, and the causes of such behavior. For instance, in all but one scheduling session involving a study with a semester or less of coursework, CBR failed. It failed a total of three times and needed to fill two to three gaps in its schedule four times, as shown in Figure 5.1.

| Student | Number of courses completed | Number of courses CBR got | Number of courses CBR Needed | Match to Plan of Study Based (In percentages) |
|---|---|---|---|---|
| A | 0 | 0 | 5 | 100 |
| B | 5 | 0 | 5 | 100 |
| C | 5 | 0 | 5 | 100 |
| D | 5 | 2 | 3 | 60 |
| E | 6 | 2 | 3 | 60 |
| F | 13 | 3 | 2 | 60 |
| G | 17 | 3 | 2 | 100 |

Figure 5.1: Number of courses completed and amount of success of CBR for each student

However, as the student progressed through their studies (i.e. acquired more hours of completed coursework), CBR had more success. In fact, the student with the most recorded coursework, seventeen courses, had the greatest success. That student's schedule needed to fill only two gaps and yet still managed to be one hundred percent similar to the plan of study based schedule.

It is possible that this high similarity is due to the fact that as a student progresses, there are naturally fewer courses left to take to complete their degree.

It is also possible that as more students "graduate", creating more completed records, CBR will become more successful.

# Chapter 6

## Conclusions

The main goal of this experiment was to observe the user experience to a virtual counseling environment. At the end of the environment, data seems to support that such a system would aid existing advising procedures. Emphasis should be placed on the word "aid" as it was also shown that not every student would use such a system. However, based on the percentage that ranked its possible reuse highly, such a system would lower the amount of students necessary to have a scheduled advising session with an advisor. This would decrease advisor workload along with allowing them more time to focus on students with more difficult problems.

Furthermore, advising software that presents multiple options in course schedules seems necessary. As the quality of the schedule cannot be measured and greatly depends upon individual preferences along with course offerings, it is important to give a student more than one option.

Therefore, AVA and possibly other advising systems used as a tool to assist advisors, but not replace them, can be an effective solution for advising the ever-growing student population.

# Chapter 7

## Future Work

AVA was implemented such that it could be tested. However, in the future a fully functioning version would be necessary. These additions would include an area that allows faculty and staff to input current plans of study, course specifics, and student information. Also, authorized users would need the ability to alter such information.

Also, for the purpose of testing, a dummy database of student information was used. To be fully functioning as designed, AVA would need to be connected to the databases that hold student records, as Oasis does at Auburn University. This would minimize the effort placed on the advisors in the amount of data entry they would have to do.

While security seems like it would be an issue, it is important to remember that the student login portion was meant to mimic existing registration software. If AVA were to be fully implemented as intended, security measures for the login would not be necessary as they would already exist in these systems.

Additionally, if AVA were to be running alongside that registration software and students were allowed to register from within AVA (for instance, once you have finished the three schedules, you could simply click one to register for), AVA would also need to be able to update the student-courses table within the database. It would also be necessary to have AVA handle conflicts of times within those schedules in order to register from within it.

However, if AVA is used to register for classes, or to go as far as determining course conflicts, it would also be beneficial to add to the interview process, possibly using speech. This could be as little as asking the user which section they would prefer, to hopefully

allow for some conflict resolution, or as far as asking the user if they prefer morning or afternoon classes, how they plan to travel between classes, which professor they prefer, how long would they like to allow for lunch breaks, and how many days they would like to take classes per week. This type of interview might even result in a more enjoyable semester for the student as well as a higher attendance rate.

An easy feature to implement in the future would be a pop up box of information. By putting a reference address in the database along with course information, a JavaScript pop up could be implemented housing information on times, locations, professors, and even a brief description of each course. This also requires more information being input into the database.

Last, but certainly not least, voice identification would make this system really stand out. If users were able to authenticate themselves by speaking it would allow the system to skip a step in the advising process. It would also lend itself to a more familiar feel, as it would appear the virtual advisor truly remembered the student without needing a login and password to "remind" them.

## Bibliography


[1] Binkerd, C. (2004). "New Approaches to Advising and Mentoring in Science and Technology". *Consortium for Computing Sciences in Colleges*: 218- 224.

[2] Carlson, E.D. (1978). "An Approach for Designing Decision Support Systems." *ACM SIGMIS Database* 10(3): 3-15.

[3] Carroll, J. and G. Chappell (1996). "An Intelligent Universal Advisor." *Proceedings of the 1996 ACM symposium on Applied Computing*: 105-109.

[4] Chubb, D. W. J. (1984). "Knowledge Engineering Problems during Expert System Development." *ACM SIGSIM Simulation Digest* 15(3): 5-9.

[5] Hage, J. (1993). "Monological Reason-Based Logic." *Proceedings of the 4rd International Conference on Artificial Intelligence and Law ICAIL '93*: 30-39.

[6] Hudson, William B. (1992). "The Evolution of Academic Advising in a Period of Change." *Proceedings of 1992 Frontiers in Education Conference*: 173-176.

[7] Kowalski, D. (1991). "Case-Based Reasoning and the Deep Structure Approach to Knowledge Representation." *Proceedings of the 3rd International Conference on Artificial Intelligence and Law ICAIL '91*: 21-30.

[8] Marques, O., X. Ding, et al. (2001). "Design and development of a Web-based academic advising system." *Frontiers in Eductation Conference*, 2001. 31st Annual 3: 6-10.

[9] Murray, W. S. and L. A. LeBlanc (1995). "A Decision Support System for Academic Advising." *Proceedings of 1995 ACM Symposium on Applied Computing*: 22-26.

[10] Pan, L. L. (1991). "An Expert system for academic advising." *Technology Management: the New International Language*: 590.

[11] Siegfried, R. M., A. M. Wittenstein, et al. (2003). "An Automated Advising System For Course Selection and Scheduling." *Journal of Computing Sciences in Colleges* 18(3): 17-25.

[12] Timmreck, E. M. (1968). "ADVISER - a program which advises students on courses." *Proceedings of 1968 23rd ACM National Conference*: 535-553.

[13] Whitson, G. M., C. Wu, et al. (1990). "Using an artificial neural system to determine the knowledge based of an expert system." *Proceedings of the 1990 ACM SIGSMALL/PC Symposium on Small Systems*: 268-270.

[14] Woolston, Donald C. (1992), "Improving Undergraduate Academic Advising in Engineering: It's Not Rocket Science." *32nd ASEE/ISEE Frontiers in Education Conference*:S2C-2 - SC2-4.

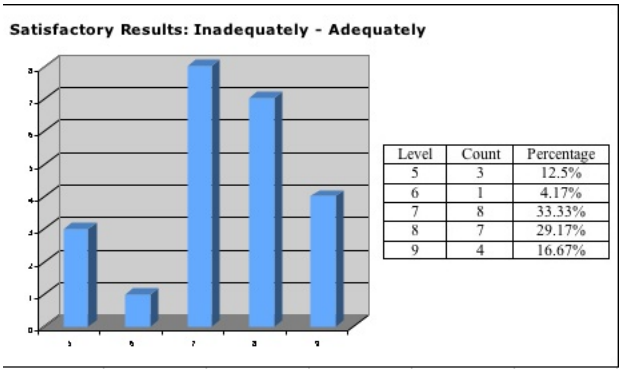
[15] Wysk, R. B. (1990). "Expert systems in the context of decision support related interventions." *Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems*: 475-490.

# Appendices

# Appendix A

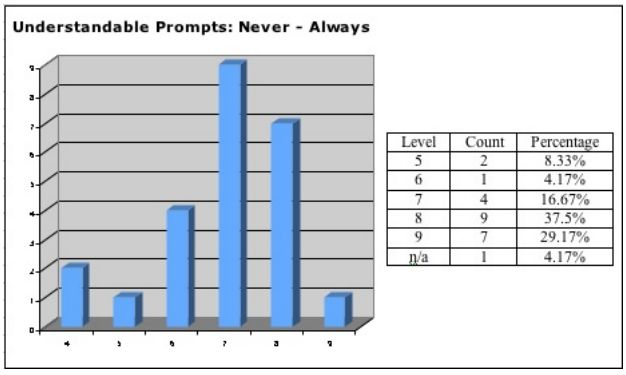
## Frequency of Responses

The following figures depict the frequency of responses for each question on the post-questionnaire.



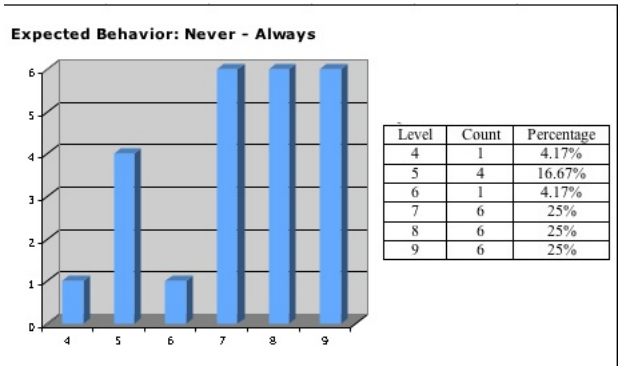

| Level | Count | Percentage |
|---|---|---|
| 5 | 3 | 12.5% |
| 6 | 1 | 4.17% |
| 7 | 8 | 33.33% |
| 8 | 7 | 29.17% |
| 9 | 4 | 16.67% |

Figure A.1: Frequencies for question 3



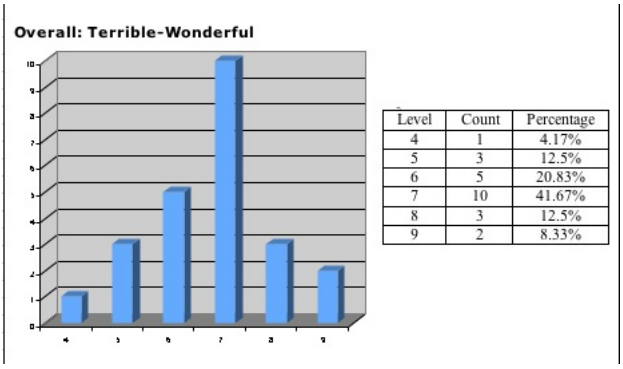

| Level | Count | Percentage |
|---|---|---|
| 5 | 2 | 8.33% |
| 6 | 1 | 4.17% |
| 7 | 4 | 16.67% |
| 8 | 9 | 37.5% |
| 9 | 7 | 29.17% |
| n/a | 1 | 4.17% |

Figure A.2: Frequencies for question 4



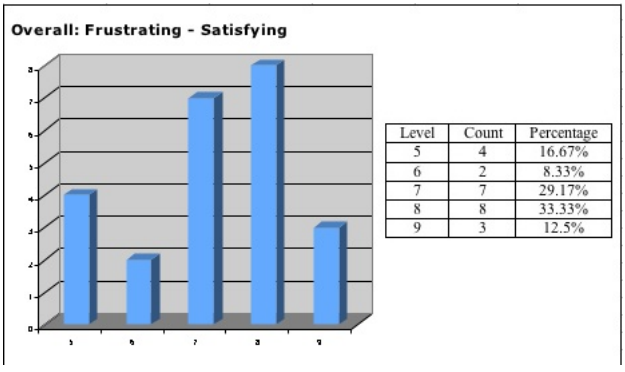

| Level | Count | Percentage |
|---|---|---|
| 4 | 1 | 4.17% |
| 5 | 4 | 16.67% |
| 6 | 1 | 4.17% |
| 7 | 6 | 25% |
| 8 | 6 | 25% |
| 9 | 6 | 25% |

Figure A.3: Frequencies for question 5



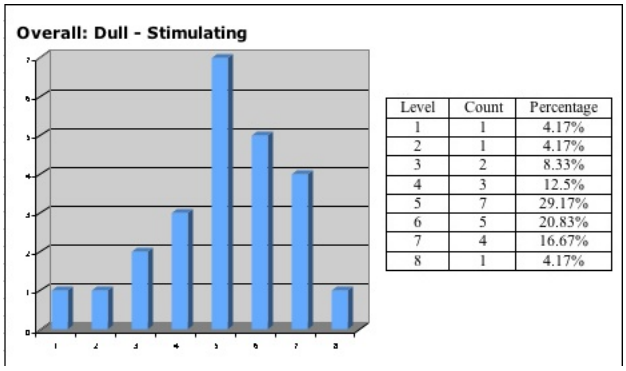

| Level | Count | Percentage |
|---|---|---|
| 4 | 1 | 4.17% |
| 5 | 3 | 12.5% |
| 6 | 5 | 20.83% |
| 7 | 10 | 41.67% |
| 8 | 3 | 12.5% |
| 9 | 2 | 8.33% |

Figure A.4: Frequencies for question 6.1



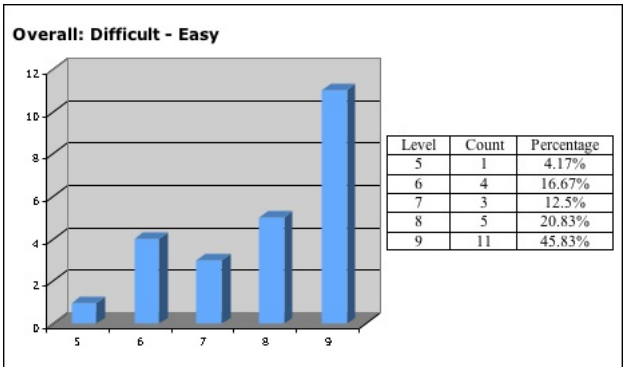

| Level | Count | Percentage |
|---|---|---|
| 5 | 4 | 16.67% |
| 6 | 2 | 8.33% |
| 7 | 7 | 29.17% |
| 8 | 8 | 33.33% |
| 9 | 3 | 12.5% |

Figure A.5: Frequencies for question 6.2



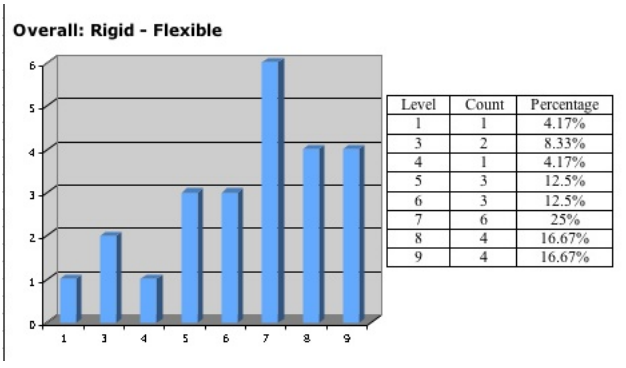

| Level | Count | Percentage |
|---|---|---|
| 1 | 1 | 4.17% |
| 2 | 1 | 4.17% |
| 3 | 2 | 8.33% |
| 4 | 3 | 12.5% |
| 5 | 7 | 29.17% |
| 6 | 5 | 20.83% |
| 7 | 4 | 16.67% |
| 8 | 1 | 4.17% |

Figure A.6: Frequencies for question 6.3



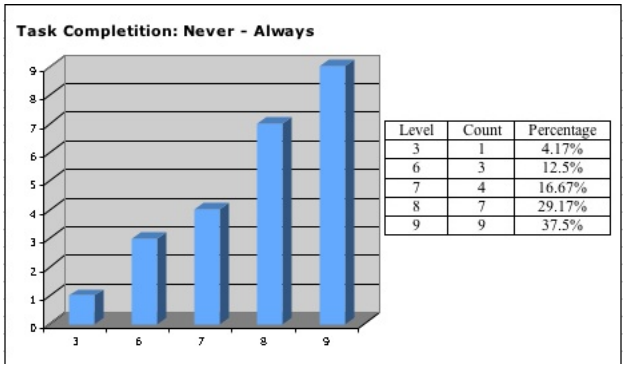

| Level | Count | Percentage |
|---|---|---|
| 5 | 1 | 4.17% |
| 6 | 4 | 16.67% |
| 7 | 3 | 12.5% |
| 8 | 5 | 20.83% |
| 9 | 11 | 45.83% |

Figure A.7: Frequencies for question 6.4



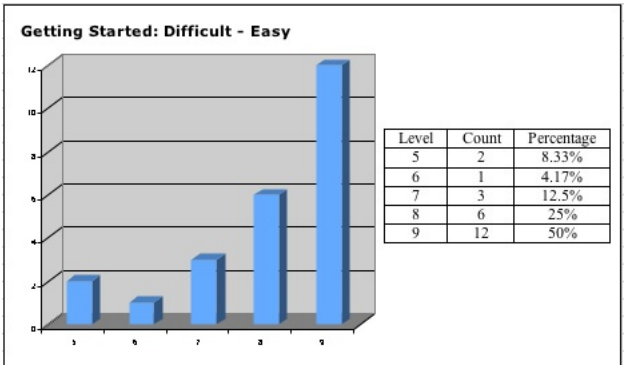

| Level | Count | Percentage |
|---|---|---|
| 1 | 1 | 4.17% |
| 3 | 2 | 8.33% |
| 4 | 1 | 4.17% |
| 5 | 3 | 12.5% |
| 6 | 3 | 12.5% |
| 7 | 6 | 25% |
| 8 | 4 | 16.67% |
| 9 | 4 | 16.67% |

Figure A.8: Frequencies for question 6.5



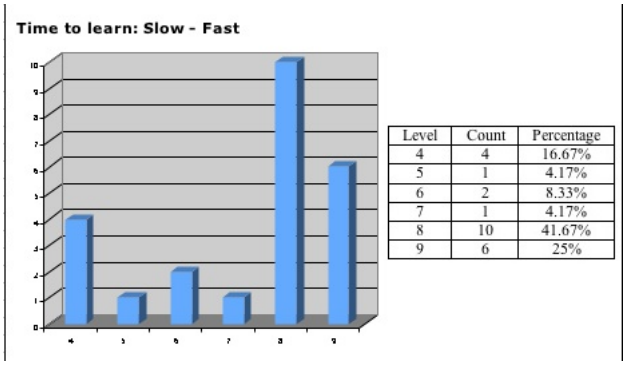

| Level | Count | Percentage |
|---|---|---|
| 3 | 1 | 4.17% |
| 6 | 3 | 12.5% |
| 7 | 4 | 16.67% |
| 8 | 7 | 29.17% |
| 9 | 9 | 37.5% |

Figure A.9: Frequencies for question 8



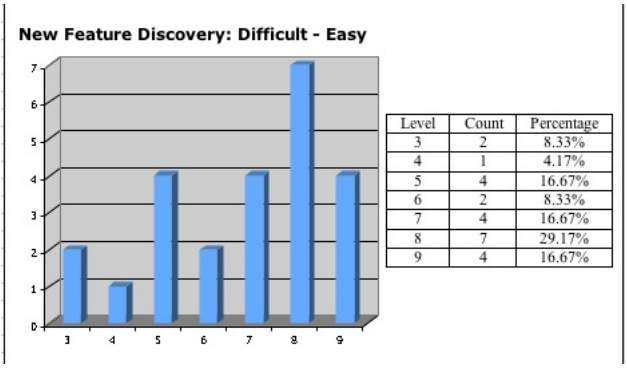

| Level | Count | Percentage |
|---|---|---|
| 5 | 2 | 8.33% |
| 6 | 1 | 4.17% |
| 7 | 3 | 12.5% |
| 8 | 6 | 25% |
| 9 | 12 | 50% |

Figure A.10: Frequencies for question 9.1



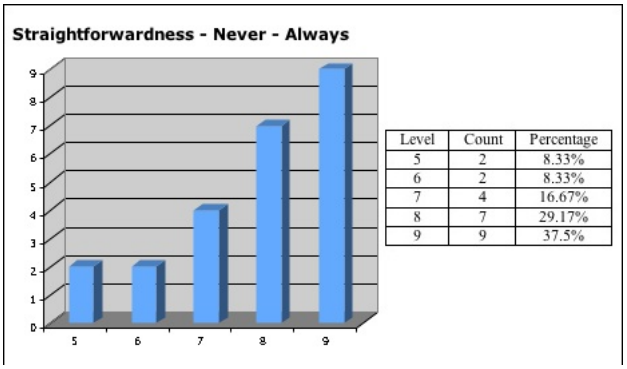

| Level | Count | Percentage |
|---|---|---|
| 4 | 4 | 16.67% |
| 5 | 1 | 4.17% |
| 6 | 2 | 8.33% |
| 7 | 1 | 4.17% |
| 8 | 10 | 41.67% |
| 9 | 6 | 25% |

Figure A.11: Frequencies for question 9.1.1



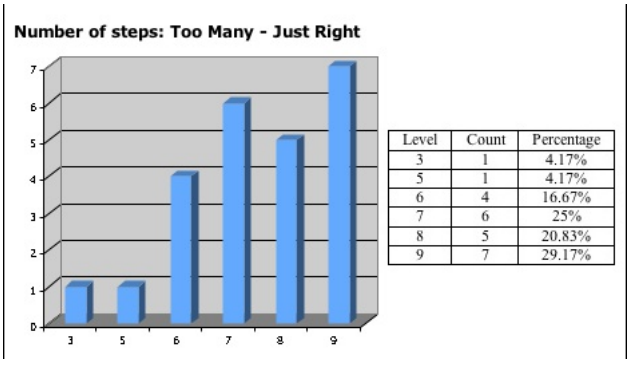

| Level | Count | Percentage |
|---|---|---|
| 3 | 2 | 8.33% |
| 4 | 1 | 4.17% |
| 5 | 4 | 16.67% |
| 6 | 2 | 8.33% |
| 7 | 4 | 16.67% |
| 8 | 7 | 29.17% |
| 9 | 4 | 16.67% |

Figure A.12: Frequencies for question 9.2



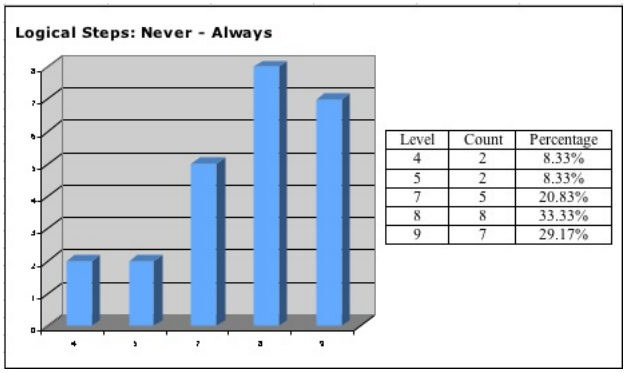

| Level | Count | Percentage |
|---|---|---|
| 5 | 2 | 8.33% |
| 6 | 2 | 8.33% |
| 7 | 4 | 16.67% |
| 8 | 7 | 29.17% |
| 9 | 9 | 37.5% |

Figure A.13: Frequencies for question 9.3



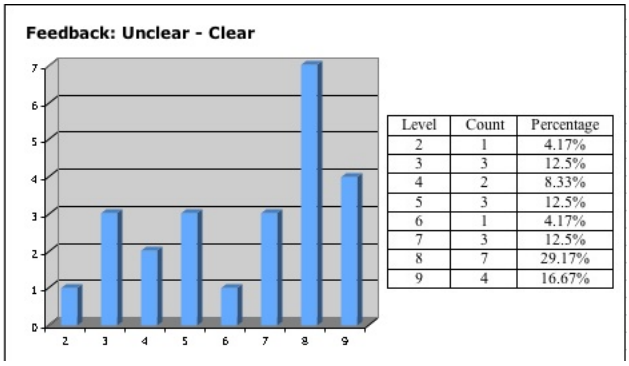

| Level | Count | Percentage |
|---|---|---|
| 3 | 1 | 4.17% |
| 5 | 1 | 4.17% |
| 6 | 4 | 16.67% |
| 7 | 6 | 25% |
| 8 | 5 | 20.83% |
| 9 | 7 | 29.17% |

Figure A.14: Frequencies for question 9.3.1



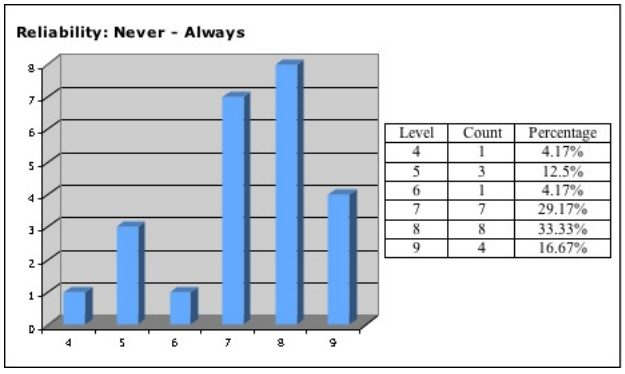

| Level | Count | Percentage |
|---|---|---|
| 4 | 2 | 8.33% |
| 5 | 2 | 8.33% |
| 7 | 5 | 20.83% |
| 8 | 8 | 33.33% |
| 9 | 7 | 29.17% |

Figure A.15: Frequencies for question 9.3.2



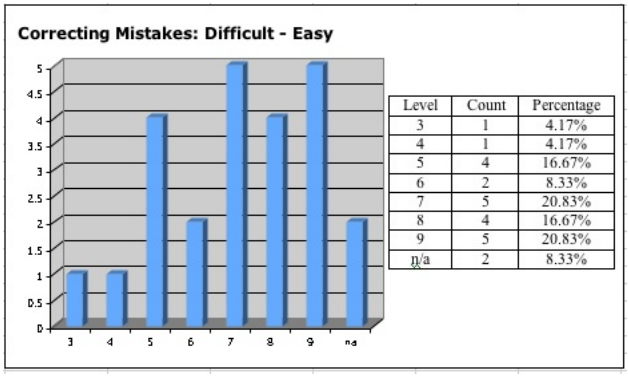

| Level | Count | Percentage |
|---|---|---|
| 2 | 1 | 4.17% |
| 3 | 3 | 12.5% |
| 4 | 2 | 8.33% |
| 5 | 3 | 12.5% |
| 6 | 1 | 4.17% |
| 7 | 3 | 12.5% |
| 8 | 7 | 29.17% |
| 9 | 4 | 16.67% |

Figure A.16: Frequencies for question 9.4



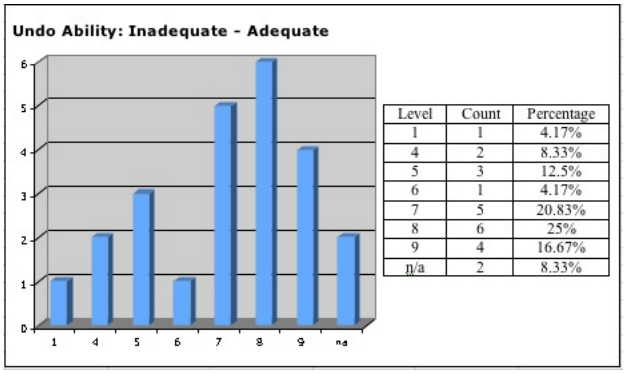

| Level | Count | Percentage |
|---|---|---|
| 4 | 1 | 4.17% |
| 5 | 3 | 12.5% |
| 6 | 1 | 4.17% |
| 7 | 7 | 29.17% |
| 8 | 8 | 33.33% |
| 9 | 4 | 16.67% |

Figure A.17: Frequencies for question 10.1



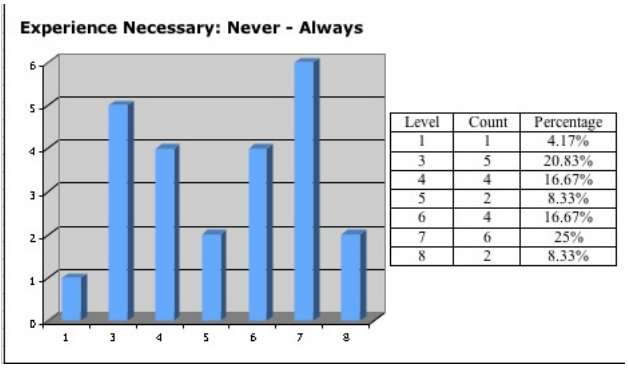

| Level | Count | Percentage |
|---|---|---|
| 3 | 1 | 4.17% |
| 4 | 1 | 4.17% |
| 5 | 4 | 16.67% |
| 6 | 2 | 8.33% |
| 7 | 5 | 20.83% |
| 8 | 4 | 16.67% |
| 9 | 5 | 20.83% |
| n/a | 2 | 8.33% |

Figure A.18: Frequencies for question 10.2



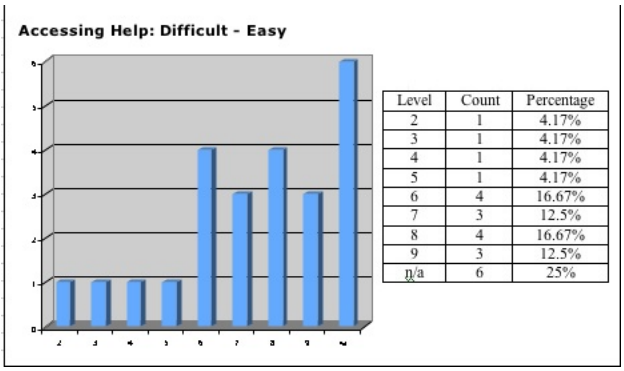

| Level | Count | Percentage |
|---|---|---|
| 1 | 1 | 4.17% |
| 4 | 2 | 8.33% |
| 5 | 3 | 12.5% |
| 6 | 1 | 4.17% |
| 7 | 5 | 20.83% |
| 8 | 6 | 25% |
| 9 | 4 | 16.67% |
| n/a | 2 | 8.33% |

Figure A.19: Frequencies for question 10.3



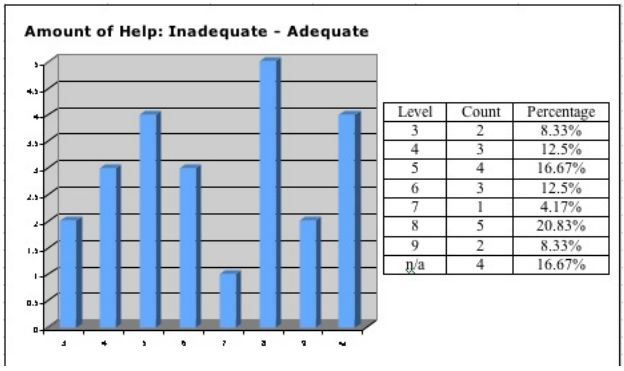

| Level | Count | Percentage |
|---|---|---|
| 1 | 1 | 4.17% |
| 3 | 5 | 20.83% |
| 4 | 4 | 16.67% |
| 5 | 2 | 8.33% |
| 6 | 4 | 16.67% |
| 7 | 6 | 25% |
| 8 | 2 | 8.33% |

Figure A.20: Frequencies for question 10.4



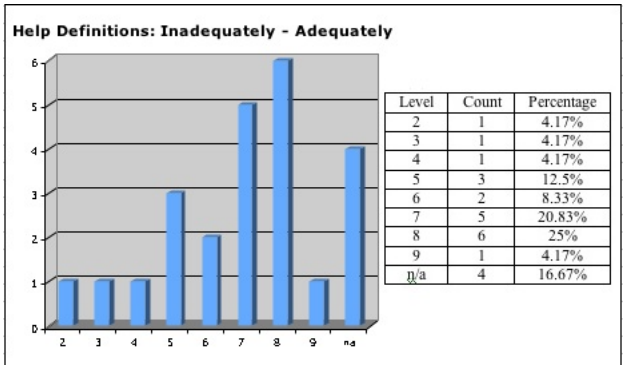

| Level | Count | Percentage |
|---|---|---|
| 2 | 1 | 4.17% |
| 3 | 1 | 4.17% |
| 4 | 1 | 4.17% |
| 5 | 1 | 4.17% |
| 6 | 4 | 16.67% |
| 7 | 3 | 12.5% |
| 8 | 4 | 16.67% |
| 9 | 3 | 12.5% |
| n/a | 6 | 25% |

Figure A.21: Frequencies for question 11.1



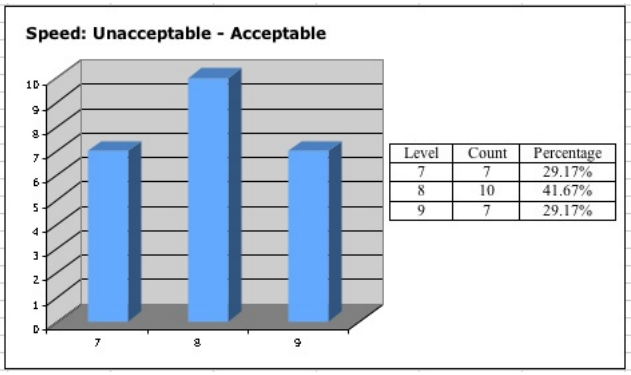

| Level | Count | Percentage |
|---|---|---|
| 3 | 2 | 8.33% |
| 4 | 3 | 12.5% |
| 5 | 4 | 16.67% |
| 6 | 3 | 12.5% |
| 7 | 1 | 4.17% |
| 8 | 5 | 20.83% |
| 9 | 2 | 8.33% |
| n/a | 4 | 16.67% |

Figure A.22: Frequencies for question 11.1.1



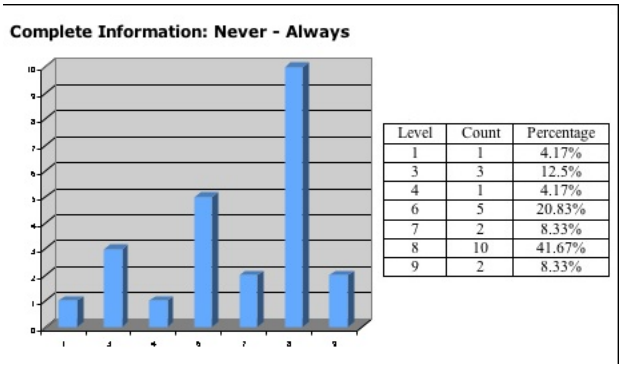

| Level | Count | Percentage |
|---|---|---|
| 2 | 1 | 4.17% |
| 3 | 1 | 4.17% |
| 4 | 1 | 4.17% |
| 5 | 3 | 12.5% |
| 6 | 2 | 8.33% |
| 7 | 5 | 20.83% |
| 8 | 6 | 25% |
| 9 | 1 | 4.17% |
| n/a | 4 | 16.67% |

Figure A.23: Frequencies for question 11.1.2



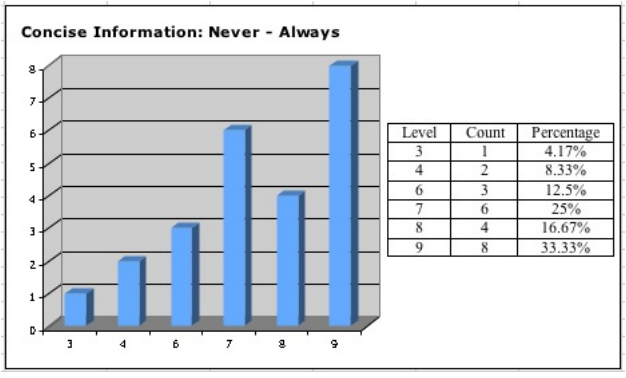

| Level | Count | Percentage |
|---|---|---|
| 7 | 7 | 29.17% |
| 8 | 10 | 41.67% |
| 9 | 7 | 29.17% |

Figure A.24: Frequencies for question 12.1



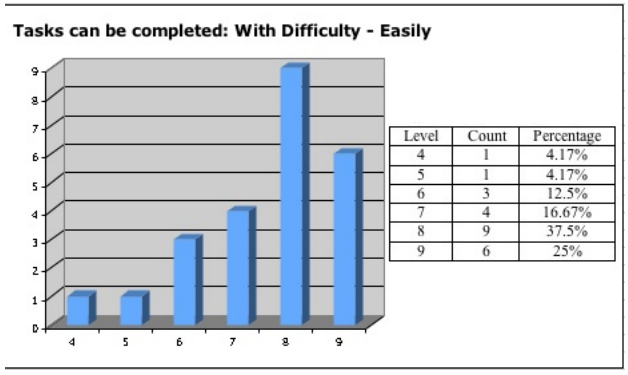

| Level | Count | Percentage |
|---|---|---|
| 1 | 1 | 4.17% |
| 3 | 3 | 12.5% |
| 4 | 1 | 4.17% |
| 6 | 5 | 20.83% |
| 7 | 2 | 8.33% |
| 8 | 10 | 41.67% |
| 9 | 2 | 8.33% |

Figure A.25: Frequencies for question 12.2



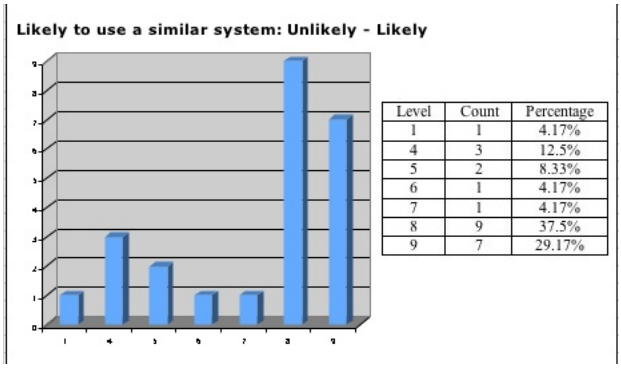

| Level | Count | Percentage |
|---|---|---|
| 3 | 1 | 4.17% |
| 4 | 2 | 8.33% |
| 6 | 3 | 12.5% |
| 7 | 6 | 25% |
| 8 | 4 | 16.67% |
| 9 | 8 | 33.33% |

Figure A.26: Frequencies for question 12.2.1



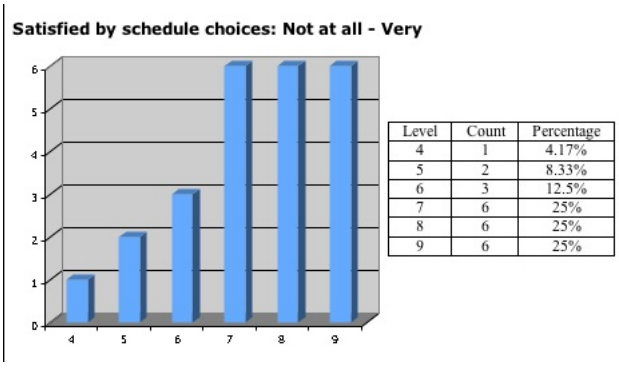

| Level | Count | Percentage |
|---|---|---|
| 4 | 1 | 4.17% |
| 5 | 1 | 4.17% |
| 6 | 3 | 12.5% |
| 7 | 4 | 16.67% |
| 8 | 9 | 37.5% |
| 9 | 6 | 25% |

Figure A.27: Frequencies for question 12.3




| Level | Count | Percentage |
|---|---|---|
| 1 | 1 | 4.17% |
| 4 | 3 | 12.5% |
| 5 | 2 | 8.33% |
| 6 | 1 | 4.17% |
| 7 | 1 | 4.17% |
| 8 | 9 | 37.5% |
| 9 | 7 | 29.17% |

Figure A.28: Frequencies for question 13




| Level | Count | Percentage |
|---|---|---|
| 4 | 1 | 4.17% |
| 5 | 2 | 8.33% |
| 6 | 3 | 12.5% |
| 7 | 6 | 25% |
| 8 | 6 | 25% |
| 9 | 6 | 25% |

Figure A.29: Frequencies for question 14

## Appendix B
## Statistics of Responses

The following tables depict the mean, median, maximum, minimum and standard deviation of responses to the questions in the post-questionnaire.

| Satisfactory results: inadequately - adequately | |
|---|---|
| mean | 7.35 |
| median | 7 |
| standard deviation | 1.23 |
| max | 9 |
| min | 5 |

Figure B.1: Statistics for question 3

| Understandable Prompts: never - always | |
|---|---|
| mean | 8 |
| median | 8 |
| standard deviation | 1 |
| max | 9 |
| min | 5 |

Figure B.2: Statistics for question 4

| Expected Behavior: never - always | |
|---|---|
| mean | 7 |
| median | 8 |
| standard deviation | 2 |
| max | 9 |
| min | 4 |

Figure B.3: Statistics for question 5

| Bi-polar Scale | mean | median | standard deviation | max | min |
|---|---|---|---|---|---|
| terrible-wonderful | 6.7 | 7 | 1.2 | 9 | 4 |
| frustrating - satisfying | 7.2 | 7 | 1.3 | 9 | 5 |
| dull - stimulating | 5.1 | 5 | 1.7 | 8 | 1 |
| difficult - easy | 7.9 | 8 | 1.3 | 9 | 5 |
| rigid - flexible | 6.4 | 7 | 2.1 | 9 | 1 |

Figure B.4: Statistics for question 6

| Task Completion: never - always | |
|---|---|
| mean | 8 |
| median | 8 |
| standard deviation | 1 |
| max | 9 |
| min | 3 |

Figure B.5: Statistics for question 8

| Bi-polar Scale | mean | median | standard deviation | max | min |
|---|---|---|---|---|---|
| Getting Started: difficult - easy | 8 | 8.5 | 1.3 | 9 | 5 |
| learning curve: slow - fast | 7.25 | 8 | 1.8 | 9 | 4 |
| new feature discovery: difficult - easy | 6.8 | 7 | 1.8 | 9 | 3 |
| straight-forwardness: never - always | 7.8 | 8 | 1.3 | 9 | 5 |
| number of steps: too many - just right | 7.39 | 7.5 | 1.53 | 9 | 3 |
| logical steps: never - always | 7.5 | 8 | 1.56 | 9 | 4 |
| clear feedback: unclear - clear | 6.4 | 7 | 2.3 | 9 | 2 |

Figure B.6: Statistics for question 9

| Bi-polar Scale | mean | median | standard deviation | max | min |
|---|---|---|---|---|---|
| Reliability: never - always | 7.25 | 7.5 | 1.39 | 9 | 4 |
| Mistake Correction: difficult - easy | 6.86 | 7 | 1.78 | 9 | 3 |
| Undo Ability: inadequate - adequate | 6.77 | 7 | 2.05 | 9 | 1 |
| Experience needed: never - always | 5.17 | 5.5 | 1.93 | 8 | 1 |

Figure B.7: Statistics for question 10

| Bi-polar Scale | mean | median | standard deviation | max | min |
|---|---|---|---|---|---|
| Help access: difficult - easy | 6.56 | 7 | 2.04 | 9 | 2 |
| Amount of help: inadequate - adequate | 6.05 | 6 | 1.96 | 9 | 3 |
| Help definitions: inadequate - adequate | 6.04 | 7 | 1.88 | 9 | 2 |

Figure B.8: Statistics for question 11

| Bi-polar Scale | mean | median | standard deviation | max | min |
|---|---|---|---|---|---|
| Speed: unacceptable - acceptable | 8 | 8 | 0.78 | 9 | 7 |
| Complete information: never - always | 6.5 | 7.5 | 2.19 | 9 | 1 |
| Concise Information: never - always | 7.29 | 7.5 | 1.76 | 9 | 3 |
| Ability to complete tasks: with difficulty - easy | 7.54 | 8 | 1.35 | 9 | 4 |

Figure B.9: Statistics for question 12

| Likely to use a similar system? Unlikely - likely | |
|---|---|
| mean | 7 |
| median | 8 |
| standard deviation | 2 |
| max | 9 |
| min | 1 |

Figure B.10: Statistics for question 13

| Satisfied with choices: not at all - very | |
|---|---|
| mean | 7 |
| median | 8 |
| standard deviation | 1 |
| max | 9 |
| min | 4 |

Figure B.11: Statistics for question 14

## Appendix C
## Information Sheet

AVA Study Directions:

1) Go to the site:
http://www.auburn.edu/~noblek/thesis/ava
2) Enter in one of the following SSNs (without dashes):

123-45-6789
111-11-1111
555-55-5555
300-00-0000
101-01-0101
202-02-0202
151-51-5151

3) For any of the SSNs listed, the password is "secret", enter it and continue to the next page
4) Review this page, as directed, click on any of the links you wish, then choose to be advised for Fall 2007, and continue to the next page.
5) Review this page, close the browser window, and then fill out the questionnaire.

Thank you for your time.

# Appendix D
## Post-Questionnaire

***Please try to answer the following questions. Please check one answer for each question.***

1. What is your gender?

   (1) Female (2) Male

2. Is English your native language or your second language?

   (1) Native language (2) Second language

3. To what extent were your query results presented to your satisfaction?

   Inadequately: 1 2 3 4 5 6 7 8 9 :Adequately NA

4. To what extent did you know what to say in response to the system's prompts?

   Never: 1 2 3 4 5 6 7 8 9 :Always NA

5. To what extent did the system behave the way you expected?

   Never: 1 2 3 4 5 6 7 8 9 :Always NA

6. Which most appropriately reflect your impression of using this system as a whole?

   **Overall reactions to the system:**

   6.1 Terrible: 1 2 3 4 5 6 7 8 9 :Wonderful NA

   6.2 Frustrating: 1 2 3 4 5 6 7 8 9 : Satisfying NA

   6.3 Dull: 1 2 3 4 5 6 7 8 9 : Stimulating NA

   6.4 Difficult: 1 2 3 4 5 6 7 8 9 :Easy NA

   6.5 Rigid: 1 2 3 4 5 6 7 8 9 : Flexible NA

8. Did you complete the task and get the information you needed?

   Never: 1 2 3 4 5 6 7 8 9 :Always

9. Which most appropriately reflect your impressions of using this system'

   Getting Started

   9.1 Difficult: 1 2 3 4 5 6 7 8 9 :Easy NA

   Time to learn to use the system

   9.1.1 Slow: 1 2 3 4 5 6 7 8 9 :Fast NA

   Discovering new features

   9.2 Difficult: 1 2 3 4 5 6 7 8 9 :Easy NA

   Tasks can be performed in a straight-forward manner

   9.3 Never: 1 2 3 4 5 6 7 8 9 :Always NA

   Number of steps per task

   9.3.1 Too Many: 1 2 3 4 5 6 7 8 9 : Just right NA

   Steps to complete the task follow a logical sequence

   9.3.2 Never: 1 2 3 4 5 6 7 8 9 :Always NA

   Feedback on the completion of the steps

   9.4 Unclear: 1 2 3 4 5 6 7 8 9 :Clear NA

10. Which most appropriately reflect your impressions of using this system

    The system is reliable

    10.1 Never: 1 2 3 4 5 6 7 8 9 :Always NA

    Correcting your mistakes

    10.2 Difficult: 1 2 3 4 5 6 7 8 9 :Easy NA

Ability to undo operations

10.3 Inadequate: 1 2 3 4 5 6 7 8 9 :Adequate NA

Ease of operation depends on your level of experience

10.4 Never: 1 2 3 4 5 6 7 8 9 :Always NA

11. Which most appropriately reflect your impressions of using this system?

Accessing help messages

11.1 Difficult: 1 2 3 4 5 6 7 8 9 :Easy NA

Amount of help given

11.1.1 Inadequate: 1 2 3 4 5 6 7 8 9 :Adequate NA

Help defines specific aspects of the system

11.1.2 Inadequately: 1 2 3 4 5 6 7 8 9 :Adequately NA

12. Which most appropriately reflect your impressions of using this system?

The speed of presentation was

12.1 Unacceptable: 1 2 3 4 5 6 7 8 9 :Acceptable NA

Information for specific aspects of the system were complete and informative

12.2 Never: 1 2 3 4 5 6 7 8 9 :Always NA

Information was concise and to the point

12.2.1 Never: 1 2 3 4 5 6 7 8 9 :Always NA

Tasks can be completed

12.3 With Difficulty: 1 2 3 4 5 6 7 8 9 :Easily NA

13. After using this system, how likely are you to use it or a similar system again?

13.1 Unlikely: 1 2 3 4 5 6 7 8 9 :Likely NA

14. To what extent would you be satisfied by the schedule choices the system gave you?

Not at all 1 2 3 4 5 6 7 8 9 Very

21. Please provide any additional comments that you have about the system:

***Note: After the experiment, please don't mention this experiment to any other students who will do this experiment later!***

***Thank vou verv much!***