The OSCAR Redesign Project Part 1 Understanding the Problem CS3750 User Interface Design

> Team: The Grouches Jack Gruendler Carlos Rios Jared Salzmann Jason Tongco

# Table of Contents

Introduction	3
Requirements Summary	3
Methodology	4
Constraints	5
User Characteristics	5
Task Analysis	6
Hierarchical Task Decomposition	7
Diagrams of Workflows or Processes	10
Object Model	11
Environment Issues	11
Usage Scenarios	11
Current UI Critique	11
Usability Goals	12
Implications	12
Reflections	13
Appendix A	13
Appendix B	14
Appendix C	15
Appendix D	

### Introduction

This OSCAR redesign is, first and foremost, not an overhaul of the backend. Such changes are beyond the scope of this project. This redesign is focused upon the user experience and placing within easy reach of the user the most-used or most-requested functions. It is meant to be a reflection of the habits and needs of its primary users: registering students.

The current frontend for OSCAR is serviceable, but it has become limited. The number of variety of classes offered by Georgia Tech has gone up, but the interface has not kept up. Searching is rudimentary at best. Navigation is limited to forward only because of POST DATA issues. Services such as permit and override requests, tuition payments, and prerequisites are hidden away and require significant navigation.

New features we would like to implement are integrated maps to help orient freshmen to campus locations; one-click or few-click access to overrides and permits; and pages dedicated to showing a class' pertinent information such as prerequisites, required texts, average instructor GPA (pulled from SGA Course Critique), and average class GPA. Other features to be considered are saved searches to help ease repeat searching, a navigation model based upon opening new windows so as to present necessary information without hindering the class search process, and an easy-to-access checklist of required classes.

This redesign of OSCAR is not meant to be a rebuke of what the previous designers have envisioned, but rather an evolution of OSCAR based upon the needs and wants of its users.

#### **Requirements Summary**

First and foremost, the redesigned OSCAR should have easy access to the functions students use frequently. Specifically, looking up classes, registering classes, and viewing schedules should be accomplished with a minimum of mouse clicks. It should be easy to navigate for the first time without the help of FASET volunteers. It should integrate more powerful search features, such as searching by GPA, and display features, such as plotting classes on a campus map. Display of information should be better organized: the user shouldn't have to look at several different pages to find out everything about a class (in contrast to the current system, which requires the user to look at the search results, click on the CRN for the class, then click "View Catalog Entry" to see all information). It should be more customizable and intelligent, by having persistent user settings (for things like search term and campus), with reasonable defaults (such as "current semester" and "main Atlanta campus"). It should not break when pages are opened in multiple windows.

#### Methodology

In our initial research we used an online survey and conducted interviews with individuals.

The surveys were intended to provide us with some detailed information about our stakeholders. The surveys were designed to provide us with data on what class information the students were most interested in when choosing a class. They also provided us with usage statistics for different tasks associated with using OSCAR and registering for classes. All questions contained a "comment" box so that users could elaborate if they wanted to. Several respondents used this feature. The survey was a great way to gather usage statistics from a large group of individuals without being intrusive and kept us from gathering data one at a time.

Our target demographic for the surveys included freshmen through seniors, graduate, and even alumni. Freshmen were particularly important because they would give us keen insight on how new users approach class registration compared to old users. Freshmen also remember how they learned to use OSCAR and can describe frustrations new users of that system face. We also targeted students of different majors to provide a healthy sampling of a student community.

We first sent the survey out to our friends so that we could make modifications to the survey if necessary. Next, we asked our current and old professors to forward our survey onto their students. Several of them complied. We also posted our survey on the git.ads newsgroup and a few pages on Facebook. We were successful in receiving a number of survey results from our entire target demographic. In total, we received 65 survey responses. The survey results are attached to this document.

After reviewing the survey results we determined what areas needed elaboration. We intended to conduct a focus group, but the participants failed to attend. Instead, we conducted semi-structured interviews with many individuals. Some of the interviews were recorded with audio. Interviews were a great way to get the students' opinions and to have them think of features or improvements that they want to see.

We interviewed some students that were our friends and coworkers. Some students, we simply walked up to and asked if we could interview them. Our target demographic for the interviews was the same for the surveys. The data that we wanted to collect was a broader sense of how the student registers for classes and why they use the methods that they use. Several key ideas were brought forth by these interviews. The interviews with freshmen revealed common problems that were faced by students that were new to the campus and new to the college. In total we interviewed 14 students. Some of questions we asked are located in Appendix A.

# Constraints

Time, Cost, and Effort

Since our user interface will be radically different from the current user interface, months of designing, constructing, and testing will take place to ensure that it will work for the thousands of students it was created for. In addition, the workers who will build the interface may have to learn new programming languages or hire new workers who have these skills, hindering the process of finishing the user interface.

Considering the complexity of the interface, a great amount of human work hours and resources will be required; because this class is only one semester and there is so much needed to do to ensure that the user interface works correctly, the user interface will not be ready to use until much later. However, what can be accomplished this semester is create a design ready to be used in constructing it. In order to use these resources and implement the user interface, cost (i.e. worker's salary, equipment, etc.) and time (i.e. building time, trials, etc.) will come into play as constraints.

Differentiation from the Competitor

Since this is a redesign of OSCAR (the sole competitor), our design must have certain features and different organization of the information if the Georgia Institute of Technology ever wants to consider using it, otherwise the user interface cannot be used whatsoever. Since this is a registration interface catered to Georgia Tech, our user interface design will be forced compete to replace the current OSCAR interface all together and cannot co-exist with it as other products are doing (i.e. operating systems, phones etc.).

# Computer Power, Memory, and Access

Because the user interface will be more complex and require other programming languages to create it, more memory and computer equipment will be needed to make the user interface a reality. This will result in more funds, time, knowledge, and energy. Also, since this user interface involves a secured database and a component of important tasks and information (i.e. registering for classes, identity information, etc.), permission from the Georgia Institute of Technology will be required in order for it to work.

# **User Characteristics**

The primary users of this system will be Georgia Tech students. This group includes undergraduate students, graduate students, and "pre-freshmen." (Registering students who have yet to begin the first semester.)

Although it was anticipated there would be a degree of disparity in needs between each level of student (freshman, sophomore, etc) what we have noticed is that freshmen have markedly different wants and perceived needs than do upperclassmen. Where juniors and

seniors requested features such as easy override access and prerequisite information, freshman wanted easily comprehensible building locations (names instead of building numbers) and as well as a list of necessary, core classes.

It is assumed that upperclassmen have a functional to solid grasp of how OSCAR works and of its various quirks and shortfalls. Freshmen, however, usually do not have any such knowledge. Some freshmen learn to use OSCAR on their own, simply by playing with the system. Others were guided through registration by upperclassmen and there are some who know nothing of the system beyond basic registration.

#### Personas

Tim is a freshman who will be registering for the very first time. He has no knowledge whatsoever of how OSCAR works or even what it is. Registering for classes is a hassle and given the monolithic nature of the OSCAR interface, it is a daunting task. Tim will need either comprehensive, automated help or an experienced user guiding him through the registration process.

Jana is a graduating senior. After eight semesters of use, Jana is well versed in using OSCAR. She knows how to modify her schedule en masse, access the weekly schedule, pay for tuition, request overrides or permits, modify grading methods, and check final grades early. She does not need much, if any help, in doing any major tasks in OSCAR.

	Freshman	Sophomore -> senior	
Sex	Male/female	Male/female	
Physical limitation	None, but may be	None, but may be	
	handicapped	handicapped	
Educational background	High school graduate, none	High school graduate, at	
	to one semester of college	least two semesters of	
		college	
Computer/IT use	Some OSCAR exp	At least two semesters of	
		OSCAR exp	
Motivation	Needs required classes	Needs required classes,	
		permits, overrides	
attitude	OSCAR is a new	OSCAR is familiar, but	
	experience.	functional	

# **Task Analysis**

When using the system, the student is primarily concerned with three tasks: finding out information about a class, registering for classes, and checking his schedule once registered. The actions required to complete these tasks may significantly overlap; for example, a student might search for a class either to find out information about it or to register for it. However, the particular details of these actions are not inherent to the

requirements of the task. Instead, they are implementation issues, to be defined during the design of the interface.

To find out information about a class, the student must look it up. In order to do this, he must know to some extent what he is looking for: for example, he can know the name of the class, the school/major that the class is part of, the level of the class, graduation requirements the class fulfills, name of the person teaching the class, time and day the class is held, etc. The student then conveys these criteria to the system, which generates a list of classes that meet it. Finally, the student selects a class from the list.

To register for classes, the student first needs to figure out which classes to register for. There are three broad categories he might choose from: classes that are required by name for his major, classes that fulfill a particular category of requirement (e.g. technical electives), and other classes that are not necessary for the degree program but that he has a particular interest in and wants to take anyway. The student can discover which classes fit into these categories in several ways: by speaking with his academic advisor, by reading the student handbook, or by searching for classes within the system in the same manner as described above. Once the student has determined which classes he wants, he attempts to register for them by looking up each class, checking his eligibility (e.g. major restrictions, level requirements, prerequisites), finding a section that is not full and does not conflict with the rest of his schedule, and selecting that section. The student may repeat these steps several times if he has difficulty with eligibility, availability, or conflicts. Finally, the process results in a satisfactory schedule.

To check his schedule, the student needs to choose which information to display and then tell the system to display it. Choices of information may include a campus map with class locations labeled, a weekly calendar showing blocks of class meeting times, the teacher's contact information, a list of required textbooks, etc.

# **Hierarchical Task Decomposition**



**Figure 1: Searching for Classes** 



Figure 2: Registering for Classes



Figure 3: Registering for Single Class

#### **Diagrams of Workflows or Processes**

Please see Appendix B.

# **Object Model**

Please see Appendix C.

#### **Environment Issues**

Our project does not have significant environmental issues. Our user interface will be displayed in a web browser so we need to make sure it is supported by all common web browsers. Users will most likely be registering for classes in a typical home or office environment. In our survey, only 6.1% of users expressed a clear desire to register for classes on a mobile device so the environmental issues associated with mobile interfaces can be avoided.

#### **Usage Scenarios**

Searching for classes: Sammy has a list of classes he needs to take for the semester. Armed with this list, he logs into OSCAR and begins searching for classes by subject. (CS, CEE, MGMT, etc). He doesn't consider the dates or times; he only cares about the class and whether they are already full.

Modifying grading method: Regina is teetering upon the edge of a 3.5GPA so she decides to take a class on a pass/fail basis. She logs into OSCAR and finds the appropriate pages to modify the grading method. She changes the grading method for her ECON 2106 class so she doesn't have to put as much effort into that as her four other classes.

Requesting overrides: Greg needs to get into a particular section of a class because all other sections presented a scheduling conflict. He takes note of the CRN and opens up the override request page. He follows those directions and is awaiting the automated reply.

Paying tuition: Diana has all her classes lined up. Her scholarship money has come in and all that's left is to hand the money over to GT. She navigates to the tuition page and after selecting a payment option (credit cards, webcheck) she makes the payment. Diana is now depressed because she's about \$3000 poorer.

# **Current UI Critique**

The current OSCAR system is functional, but according to personal use and from interviews with current users, it's becoming outdated and a hindrance to good use. The most glaring example is probably the class search page that attempts to contain within a three-line scroll box the dozens and dozens of subjects to choose from. Essential functions such as override requests and tuition payment options are not easily accessible and in fact require a fair amount of navigation to find. The inability to use the browser's back button to navigate is grating, especially when looking at course prerequisites or catalog entries.

Our design aims to use Web 2.0 technologies to help extend the user interface. We will change the way users search for classes: instead of the strict criteria imposed in the current implementation of OSCAR, we will use a more abstract method of searching, similar to any common search engine. An example would be searching for "calc" and having returned results for calculus I, calculus II, etc. As it stands, search is limited to specific search strings, proving to be quite unsuitable for general searching.

In terms of usability, the goal is create a system which puts in easy reach the features that more experienced users need (overrides, transcripts, etc) as well as being intuitive enough for a first-time user with minimal instruction to use effectively. Given the broad audience to be sought (freshmen to seniors), a means of hiding or abstracting away functions might be needed.

	Min Time	Max Time	Learned Time
Fill out 3 search	1 min	10 min	30 seconds
filters			
Find 1 class called	10 sec	20 sec	5 sec
"Calculus 1" out of			
list of 10 classes.			
Find out which	2 min	5 min	1 min
classes out of 10			
random classes has a			
prerequisite			
Add a class to	1 min	3 min	3 seconds
worksheet			
Priorities classes in	3 min	10 min	12 seconds
worksheet			
Perform permit or	1 min	5 min	30 seconds
override request			

# **Usability Goals**

# Implications

The goal of a class registration system is to provide the most relevant information to the user in an easy to understand way. Through our research we have concluded that the new OSCAR must provide features and user interfaces for two different types of users, freshman students and non-freshman students. A third category of users includes all students.

Freshmen are new to the school and new to the OSCAR so freshmen are in need of more guidance as they are registering for classes. Unlike upperclassmen, freshmen do not know where any of the builds are located on campus and how long it takes to move across campus. This causes them to accidentally register for classes that are too far apart. Freshmen also have way to gather information about classes in general. Freshmen may not have anyone to ask "is Bio easier than Chem?" or "what do they teach in Calculus 1?" In general, freshmen are more concerned with what classes they are eligible to take than they are with choosing which classes to take. Unlike upperclassmen, our research showed that freshmen usually do not request overrides. Our research also showed that some freshman didn't even know how to apply for overrides. All these problems that freshman have is a result of their lack of experience. Freshmen are also unable to find access the knowledge or data that would displace their lack of experience. The class registration system should attempt to gather the information and show it to the user or refer the user to the location of the information.

Upperclassmen have slightly different requirements from a class registration system. In general, upperclassman don't need hand holding. Upperclassmen usually have more class options and must choose between a large numbers of classes. Seniors are mostly concerned with knowing what classes they need to take to graduate. Unlike freshmen, upperclassmen need immediate access to request an override.

There is a lot of information that all students seek when choosing their classes. Every student we interviewed said that when considering what classes to take they either look at a graduation requirements checklist or they ask their advisors. This is a strong argument for integrating a requirement checklist or degree planer integrated into the class registration system. The registrar can provide such checklists based on the major of the student.

Many students we interviewed informed us that they often receive search results that are incorrect, not relevant, or superfluous. Half of the students we talked with preferred to display huge numbers of classes and relied on their eyes to search for classes. This is a sure sign that search and the search results display need to be improved. In order to improve the quality and accuracy of search results, the search function could become fuzzy. If a user searches for "calc" the fuzzy search would understand that calc is short for calculus. A "did you mean" feature could also improve the accuracy of search results by providing post search filtering. Most students were agitated that they had to perform the same searches over and over again. Possible solutions to this problem could be saving a user's recent searches or guessing search parameters based on previous searches.

After the results are displayed features need to be added to help the user organize and read the data. Students said that many search results contained classes that they could not register for. Solutions to this problem include a color coding of results based on availability or relevance or a "hide unavailable classes" button. Dynamic interfaces such as those found on Web 2.0 websites could enhance the visual appeal and usability of the search results listing. Dynamic pages could display critical class information but quickly transform to show all the details of a class.

During our research, students often relied on outside tools. SGA's Course Critique and campus map data could be integrated into the class registration system. Other outside tools included calendar and spreadsheet applications. Some students even used pen and paper. A worksheet that provides the user with a location to organize and play with the classes they are interested in could replace these external tools. Students could have the

ability to add classes to a "worksheet" where the classes could be expressed as items on a weekly calendar or on a list sorted by priority. The student could make notes about each of the classes such as "Jon said this was easy". The worksheet is where the student would decide exactly which classes to take from a list of classes they can and would like to take.

# Reflections

The most difficult part of the project was creating this extensive report. Getting people to participate in a survey was difficult but did not take too much time. Getting people to participate in a focus group proved to be too difficult so we had to walk around campus to collect interviews. About 50% of the people we approached allowed us to proceed with the interview. Creating the survey was fairly easy but it did cost \$20 for a Survey Monkey account. If we had to do it over again, we would have started the interviews or focus groups while the survey was still going on. If we had more time, we would have liked to interview people at the Registrar to see what types of questions students ask about OSCAR. Also, people at the Registrar might be able to explain why OSCAR does things in a certain way. We would have also liked to interview advisers to see what types of questions students ask them and what data students request.

# **Appendix A: Sample interview questions**

What do you like/dislike about OSCAR? How would you change that to make it better? How could existing features of OSCAR be improved/removed? What are some features that are lacking from OSCAR? Have you used a registration/scheduling system other than OSCAR? Pros/Cons? What tools do you use to register? How do they help you? Do you use a weekly calendar to view your schedule? What about the calendar helps you organize your classes? When searching for classes, do you suffer from information overload or underload? Have you applied for an override request? If so: How could the override request process be better implemented? Should it be integrated in class search results? If not: do you know what an override request is and how to apply for one? How could classes be color coded for better info gathering? When and how did you learn to use OSCAR? What stumbling blocks where there that hindered your class registration? Can you think of a way to remove that stumbling block? What "features" slow you down when registering? How can those features be improved/removed to help you register? Are there user preferences that you want to save between sessions? Do you print out a page with all the "necessary" class information on it for the fist day of classes. What is on that page?