Kevin McIntosh, Joe Fillmore, Tammi Fuchigami, Jonathan Lee
CS 352
Evaluation Results

Empirical Evaluation: Usability Study

Empirical Test Plan

The task we evaluated is that of a student who is a Computer Science major on the Computer Systems track who has completed one term of classes at OSU and is planning the rest of their academic year. We chose this scenario as it allows us to get an idea of how intuitive both the initial planning process and the pre-planning processes are, based on the planning resources the system will provide.

We chose a male Mechanical Engineering student to perform the study on. We felt this student would be a suitable user in particular, because they have no knowledge of the existing processes the school of EECS uses for academic planning and so they won't be biased towards it. The user had to be able to grasp what courses they needed to take to advance their academic career exclusively from the resources that we provided in the interface. We also felt this student was suitable because he has what we expect to be a standard level of ability in computer use.

For the Empirical Evaluation we did a usability study using a paper prototype of our system. We chose this method of evaluation because one of our main goals is to make the academic planning process easier to learn, and the usability study provided us with some insights on how well our interface is achieving that goal. We chose the paper prototype for this, as it is fairly easy to construct while still being a viable means of testing the interface. It also allowed the user to get a somewhat interactive experience because we are using a dynamic paper prototype. We avoided using a hi-fi prototype too early on in case there are any big changes that we need to make.

How Analysis Was Performed

One of the members in our group met with the Mechanical Engineering student. The student in question is a male, with Junior Standing who is a transfer to OSU from Portland Community College. The walkthrough was done in the Kelly Engineering Center Atrium at about 5:30. As is fairly usual for that building at that time of day, there was still some activity, but most of the lounge areas were pretty sparse, most of the tables along the north side of the Atrium had between 1 and 2 people at them, and there was very little in the way of foot traffic.

For the walkthrough, the user was given the task of scheduling classes to take for winter term 2010 given that they had already completed some courses in the fall. They were not told whether they had passed these courses or not. We then had the user start at the course schedule screen. From here the user walked through adding courses to their schedule with their only reference

being from the list and tree view of course offerings. After they had completed this task for winter term, they went back to the course schedule screen, synched their plan with the transcript, and found a conflict. The user then went back into the tree view of the course offerings, and altered their plan. They finished by going back to the course selection screen and saving their academic plan.

The walkthrough was done on a dynamic paper prototype. The list and tree view of classes we provided were only 'interactive' in that the check boxes used to add courses to a user's planned schedule could be checked and unchecked. The course selection screen was interactive, in that it would display courses that the user had added in the tree and list a view of course offerings.

Both due to limitations of the paper prototype and the interface concept itself, the user required some additional instruction in order to complete tasks described above. These additional instructions were fairly minor, though did tend to bring to light issues with the interface concept.

The additional instructions were:

- From the schedule courses screen, the user was unsure what to do. A suggestion that the underlining of the uncompleted terms was important was made to the user, who figured it out from there.
- Due to a formatting issue, asterisk on the list view of courses were changed to bullet points in a few places. Since these asterisk were used to denote pre-req's that could be taken concurrently with a course, it caused a little confusion.
- The user asked if all the courses in the list and tree views were supposed to be available for that term, and if there was any way to examine the times those courses were offered. Since that was something we had overlooked in our design to that point, the user was just told to assume the tree and list views only displayed courses that would be available for the given term.
- An error in the creation of the paper prototype made the user confused about what courses they had taken and what they hadn't, when using the tree view of course offerings. To remedy this, the user was told what courses they were supposed to have taken during the fall and used the course schedule screen for reference of what they had already taken.
- When a course was added in the tree view, it was unclear what feedback would be given to the user that they had correctly added the course they wanted to their planned schedule for the given term. For the purpose of the study, cut outs of the phrases "Scheduled" and "Prereq. not completed" were used to convey feedback when appropriate. Also due to an error in the paper prototype creation, courses for which the pre-req's were not completed were not grayed out. This was handled by the aforementioned feedback.
- While there is an indication given on the tree view that courses are co-required, the user wasn't sure how to interpret the visual cue, and had to be told what the line with no arrows on the end represented.
- Certain courses (in specific, COMM 111 and COMM 114) fulfill the same requirement, and should be taken in a sort of 'either-or' capacity. The user noticed that both COMM

classes were on the tree view, and had to be told that they were only to take one or the other.
- There was no obvious way in the interface to go from either the tree or list view back to the course schedule page. The user was instructed to simply assume such a mechanism existed.

Analysis

This list of extra instructions given to the user during the usability study also (for the most part) doubles as a list of problems we found with our interface. In addition to the problems found that required direct comment, the following problems were found:

- The user spent about 5 minutes looking through the list view of courses before they noticed the "View: List Tree" component of the interface at the top of the screen, and switched from the list to the tree view of courses. Something could be done to make the existence of the two views more visible to users.
- The first action the user took after adding a course to a schedule was to ask which other courses they had already scheduled for the term. Having the ability to see that information explicitly in both the list and tree view would be useful.

Insights

From these results, we've come up with the following insights:

- The interface needs to offer a way to return to the course schedule screen from the list / tree view screens.
- We should address the issue of courses not being available every term needs to be addressed in the interface.
- We should address the issue of having multiple courses filling the same requirement (specifically, COMM 111 and COMM 114) needs to be addressed in the tree view.
- While it didn't come up here there needs to be a representation for each of the Baccalaureate Core Courses in both course offering views.
- Having a separate and explicit list of courses that the user has already scheduled for a given term in both views could be useful.
- Having more explicit feedback that a course was successfully added to the users planned schedule could be useful.
- There could be a more obvious way for the user to get to scheduling courses for a term than having them click the underlined term name.
- There could be a more obvious way of making the user aware of the different way the system has of displaying course offerings.

Empirical Evaluation Data

Notes

- Unclear on where to go from initial page
- From list view, hard to tell what to take
- No feedback on what was clicked besides check. Needs more visibility on scheduled list page
- Added CS 161
- Issue with asterisk showing as dot
- Immediately say pre-req not met for appropriate classes
- Trac found
- "That's handy, I like that"
- Do courses viewed in the list mean it's offered that term
- Grey out those not being used
- MTH 231 scheduled
- Click lines on the tree?
- PH 211 registered - no reqs brought up
- Looked for courses added that term when trying to decide to register for PH 211 and MTH 252 and PH 221
- Didn't have feedback for taken courses
- What about co-req courses (COMM 11/114)
- No back to main view button
- Not sure what to do (to get back to main view)
- Spring discussed
- Clicking save is good
- "No obvious way to go remove them"
- Click to remove button on main view
- What was removed (PH MTH 252)
- Retake MTH 251
- Picking new class not pre-req dependent (WR 121)
- Picked ECE 271

Analytical Evaluation: Heuristic Evaluation

Heuristic Evaluation Test Plan

For the analytical evaluation we used Nielsen's heuristics to evaluate our UI. Using the guidelines Nielsen presents we determined if we have a suitable UI. This evaluation highlighted areas that can be improved to allow for a better overall user experience. The specific heuristics we checked are listed below.

The tasks that we evaluated are adding new courses to a student plan and modifying courses within an existing plan. We chose these tasks because they represent the main feature of the academic planning system. The whole purpose of the system is for EECS students to be able to create and modify their academic plan while enrolled in the College of EECS. These are the tasks that students will be spending most of their time on while using our system so we want to ensure these tasks are optimized as much as possible for a pleasant/efficient user experience.

How Analysis Was Performed

Our analytical evaluation was performed using Nielson's heuristics. Two members our team each did an independent evaluation. One member evaluated the tasks of adding new courses and modifying an existing plan. The other member evaluated the task of adding new courses. The perspective taken was of an EECS student (male/female, average age 18 to mid-twenties, computer literate). The questions we asked ourselves while doing the evaluation are as follows:

- **Visibility of system status**: Is the system providing appropriate feedback at reasonable times?
- **Match between system and the real world**: Is the system providing "phrases and concepts familiar to the user, rather than system-oriented terms?" This includes real world conversations.
- **User control and freedom**: Does are system support a undo/redo system?
- **Consistency and standards**: Is the terminology that has been used consistent with the system and platform?
- **Error Prevention**: Does the design prevent a problem from occurring? If an error occurs, does the the system provide a detailed enough message that related what went wrong with the program? If the user is able to use an action that may or may not cause an error, does the system ask for conformation before allowing the user to proceed?
- **Recognition rather the Recall**: Does the interface reduce the memory load of the user by making all of the objects, actions, and options visible when applicable?
- **Flexibility and efficiency of use:** Is it possible for the system to cater to both experienced and novice users through the speed at which the task can be completed? Can the user "tailor frequent actions"?
- **Aesthetic and minimalist design**: Is all the information that is displayed intended to be there; does the displayed information on the GUI have relevant qualities?

- **Help users recognize, diagnose, and recover from errors**: Are error messages "in plain language (no codes), [do they] precisely indicate the problem, and constructively suggest a solution"?
- **Help and documentation**: If help is needed with the interface, will the user be able to find the
- right help documentation and understand all concurrent instructions.

After performing the heuristic evaluations, we brought the results together and discussed what we found. Our analysis and discussion resulted in the following:

Analysis

In analyzing the results of the two independent heuristic evaluations it became clear to us that we need to incorporate more feedback for the user. Examples being when the user saves their selected or when the user syncs his/her transcript to the plan. Another example would be showing the credit hours of each course selected. That is definitely a key piece of information students will want to see.

Both analyzers felt the system uses the correct language for the intended users and incorporates functionality the user would be familiar with (dropdowns, link, check boxes, etc). The users should be able to understand what is being communicated to them via this system and how to perform the necessary actions.

For the most part this system also provides a pretty good real world match. The course information provided in this system should match the information provided in the course catalog students are used to using. One area of improvement might be using the word synch on the transcript synch button. This maybe confusing to some as they may expect the academic planner and transcript to each sync whereas the sync is really one-sided only the academic planner will update.

User control and freedom may need to be improved upon. Currently there is not a way for students to redo/undo their changes. This feature may want to be implemented on the main screen allowing the user to reset to his/her last submit. On the course selection screens per term, a reset may not be necessary as the student will only be selecting approximately five courses per term.

The internal consistency and standards of the system are sound. It may take the user a little time to get used to the tree view course selection options since this will be a view they are not necessarily used to. Given this we may need to provide more help features on this screen. In addition, the student does not have to use this screen. They can use the list view. Having both options does provide more flexibility to the user.

The system does a good job of error prevention. It prevents the user from selecting a course if the prereq has not been planned in a prior quarter. The system prevents the user from selecting a

course they have already taken. An area of improvement would be providing an error message if the system fails a user action for some reason.

Recognition is taken advantage of in this tool. We may need to provide the course name along with the course number in the main screen to provide the user with a little more information. In addition we may want to provide hover information with the course description in-case the student still isn't sure what the course is. Overall with the information provided to the student they should be able to recognize what they need to.

Flexibility and efficiency of use is an area of improvement we can incorporate better in the main screen. Currently the student would need to go to each terms' selection screen to modify the plan. It would be more efficient for the user if they could modify their current plan on the main screen - deleting or moving courses to a new term.

Aesthetic and minimalist design concepts have been taken into account. The system only displays relevant information to the students plan. In the selection screen the quick select courses will be limited to those of the student's major/track.

We need to incorporate more help/documentation features for the users. This includes some high level instructions, error/recovery messages, hover information/help features.

Overall we felt we have a good start at the usability design of this system. This evaluation definitely highlighted areas of improvement that we had not thought of previously and plan to incorporate in our next version.

Insights

- Error messages (Immediate Interruptions) need to be present in cases where saves fail or "synch with transcript"
- There needs to be a help document describing what to do and how to do it. Our interface is not intuitive enough.
- Classes that you do not have the prerequisites for need to be greyed out aka not selectable.
- Users need to be able to delete from the main screen
- Drag and drop need to be used from the main screen
- Status bar at the top of the screen that shows how close the user is to graduation based on how many classes that they have completed.
- If synched with the transcript, grades need to be displayed on previous classes.
- There needs to be an "undo" button in the form of resetting the schedule to the start of the session for which they logged in.
- There needs to be an extra field at the bottom of the list that allows users to add in other classes. This allows the interface to be more dynamic.
- There needs to be more of a hook in using the tree. There does not seem to be enough of a hook. Maybe making it more intuitive?
- Remember the view preference of the user; each interface has a way of remembering how the user views the classes.
- Uploading a suggested transcript may have streamlined the process for initial scheduling.

Heuristic Evaluation Data 1

1st Iteration: Adding new classes to plan

Visibility of system status: Is the system providing appropriate feedback at reasonable times?

Positive Attributes:
The system provides appropriate feedback when a user selects a class to take by showing the class they have selected in the selection screen and displaying the class in the main screen once the user has saved his/her selection. The system highlights to the user the classes that have already been taken or require a prerequisite before being able to be selected.

Improvements:
On the selection screen one thing we may want to consider is making a note that selections are not saved to the plan until the user actually saves their selection. When the save button is selected we might consider some sort of indicator that selections are being saved if the save is taking over a certain amount of time. There should be a color code explaining what highlighted courses that can't be selected are (already taken, require perquisite.)

Match between system and the real world: Is the system providing "phrases and concepts familiar to the user, rather than system-oriented terms?" This includes real world conversations.

Positive Attributes:
The system does use phrases and concepts experienced EECS students will be familiar with. The course information should match the course catalog exactly – in fact be pulled from the same database if possible.

Improvements:
Major selection should not use acronyms – each major should be written out. Even though most students will know what the acronyms mean, new students may not. Spelling it out will avoid confusion. MECOP selection should provide a description (pop-up box) explain what MECOP is. Again, new students may not be familiar with the term.

User control and freedom: Does are system support a undo/redo system?

Comments:
System does not currently have an undo/redo built in. The student can just unselect or reselect courses chosen or close out of the selection screen without saving changes. Given that they can only plan one term at a time they will not be selecting many courses at a time. I feel an undo/ redo is not necessary in the selection screen.

Improvements:

The main screen may need a reset to previous submit option. This would allow the student to add classes for multiple terms and then if they decide they do not like the additions, they can reset the system to what they originally started with when they first entered the system.

Consistency and standards: Is the terminology that has been used consistent with the system and platform?

Positive Attributes:
The system is consistent between screens. Course information is the same whether you are on the main screen or one of the selection screens. Both selection screens will require the user to save courses selected. Both will allow for a search of bac core classes or electives. Links will be underlined to indicate a link.

Improvements:
May need some sort of indicator in areas we have hover help/info available.

Error Prevention: Does the design prevent a problem from occurring? If an error occurs, does the system provide a detailed enough message that related what went wrong with the program? If the user is able to use an action that may or may not cause an error, does the system ask for conformation before allowing the user to proceed?

Positive Attributes:
The student is not able to select courses already taken or courses that require a prerequisite that has not been scheduled in a previous term. If a user has a class scheduled in which they fail the prerequisite the system will highlight the course letting the user know the prereq has not been met. Student must select a Major and Track before being able to add courses to plan. Courses available to select by checking a box are limited to those in the selected Major/Track. (The student can still select other courses by doing a search. They just aren't easy select courses.)

Improvements:
An error message needs to be displayed if the save or submit feature fails. (The student will also know the save of selected courses failed if when the selected classes don't show up on the main screen.)

Recognition rather the Recall: Does the interface reduce the memory load of the user by making all of the objects, actions, and options visible when applicable?

Positive Attributes:
The system uses dropdown lists for selection of Major and Track. The system provides the user with a list of courses to select from and a search option for courses not within their Track. Descriptions of the courses are also provided in the selection screen.

Improvements:

In the hierarchy selection screen may want to provide course name with course number as well as hover info with the course description. The system should also include this for the selected courses displayed on the main screen.

Flexibility and efficiency of use: Is it possible for the system to cater to both experienced and novice users through the speed at which the task can be completed? Can the user "tailor frequent actions"?

Positive Attributes:
The system provides the student with two ways to select courses – list view or tree view.

Improvements:
The system should remember the view the student prefers (or last used) and bring that view up when the student accesses the scheduling screen. This will prevent the user from having to select their preferred view each time.

Aesthetic and minimalist design: Is all the information that is displayed intended to be there; does the displayed information on the GUI have relevant qualities?

Positive Attributes:
The system only displays relevant information. The selection screens limits the courses that are auto-populated to those that are relevant to the Major/Track the student has selected. Information displayed is enough to plan a schedule but not too much.

Help users recognize, diagnose, and recover from errors: Are error messages "in plain language (no codes), [do they] precisely indicate the problem, and constructively suggest resolution"?

Positive Attributes:
If a course can't be selected because a prerequisite is required the system provides information regarding what prereqs are required.

Improvements:
System needs to provide color code of reasons for courses that can't be selected. If the save or submit fail, need to provide the student with information regarding what to do about it – contact support.

Help and documentation: If help is needed with the interface, will the user be able to find the right help documentation and understand all concurrent instructions.

Improvements:
Provide a short list of instructions at the top of main and selection screens regarding how to add classes.

Task: Modifying plan

Visibility of system status: Is the system providing appropriate feedback at reasonable times?

Positive Attributes:
Same as adding courses.

Improvements:
Same as adding courses. On the selection screen, it may be good to indicate what courses have been modified – newly selected or deselected. On the main screen if may be good to highlight modified courses so the student can easily see what they have changed until they actually submit the plan.

Match between system and the real world: Is the system providing "phrases and concepts familiar to the user, rather than system-oriented terms?" This includes real world conversations.

Positive Attributes:
Same as adding courses

Improvements:
Same as adding courses.

User control and freedom: Does are system support a undo/redo system?

Improvements:
As mentioned with adding courses, the main screen may need a reset to previous submit option. On the main screen the system should allow students to delete or move courses from one term to another. The reset would allow them to undo any changes they had made – as long as they had submitted them yet.

Consistency and standards: Is the terminology that has been used consistent with the system and platform?

Positive Attributes:
Same as adding courses.

Improvements:
Same as adding courses.

Error Prevention: Does the design prevent a problem from occurring? If an error occurs, does the system provide a detailed enough message that related what went wrong with the program? If the user is able to use an action that may or may not cause an error, does the system ask for conformation before allowing the user to proceed?

Positive Attributes:

Same as adding courses. The system does not allow students to modify courses that have been updated based on their transcript.

Improvements:
Same as adding courses.

Recognition rather the Recall: Does the interface reduce the memory load of the user by making all of the objects, actions, and options visible when applicable?

Positive Attributes:
The main screen shows the student the courses they had previously selected in the term they planned them for.

Improvements:
Same as adding courses.

Flexibility and efficiency of use: Is it possible for the system to cater to both experienced and novice users through the speed at which the task can be completed? Can the user "tailor frequent actions"?

Improvements:
The system should allow the student to delete or move a course (drag and drop) from one term to another from the main screen. This will prevent them from having to go to the selection screen each time they make a modification.

Aesthetic and minimalist design: Is all the information that is displayed intended to be there; does the displayed information on the GUI have relevant qualities?

Positive Attributes:
The system only displays relevant information. Information displayed is enough to know what has been previously planned. The system highlights when a course has been planned but the prerequisite has not been met.

Help users recognize, diagnose, and recover from errors: Are error messages "in plain language (no codes), [do they] precisely indicate the problem, and constructively suggest resolution"?

Positive Attributes:
Same as adding courses. If a user fails a course that was a perquisite for a course scheduled later that later course will be highlighted indicating the prerequisite was not met.

Improvements:
Same as adding courses. Hover info box needs to provide student with list of required prereqs if the course is highlighted as prereq not being met.

Help and documentation: If help is needed with the interface, will the user be able to find the right help documentation and understand all concurrent instructions.

Improvements:
Provide a short list of instructions regarding how to modify course plans at the top of main and selection screens.

Heuristic Evaluation Data 2

Visibility of system status: Is the system providing appropriate feedback at reasonable times?

Since the interface's method of scheduling classes is through a student manipulating elements of complex lists and trees and placing these facets within his/her schedule, the student/user requirement for feedback becomes paramount.

This particular "feedback" within the UI is characterized by:

- Green "check marks" for whether or not the user can take the class
- Red "x's" indicating that the user cannot take that class at the specific time
- Yellow highlighting of class descriptions/labels for classes the student is taking but still has yet to complete the requirements for it
- Check boxes next to class titles that the user can take

However, there we some aspects missing that could prove to be useful in providing feedback:

- There is no indicator for how many credits the student is taking
- No notification is given after the user applies their changes to their schedule.
- No notification is given after the user uses the function: "Sync with Transcript"
- The tree feature does not re-draw out lines e.g. changes their color when the user schedules a new item in the tree option of the scheduling function.
- There is no feedback if a class is a co-requisite or not
- The physical printed schedule is the only indicator of how many classes the student has completed. A status bar displaying how much the user has completed would be useful.

Match between system and the real world: Is the system providing "phrases and concepts familiar to the user, rather than system-oriented terms?" This includes real world conversations.

This part, to the best that I can interpret, is using traditional phrases and concepts such as dragging, dropping, syncing, etc. so that the user can utilize the interface more carefully. Also, this section evaluates some of the structures that the user can manipulate.

One example of this is the "Sync with transcript". The sync option is a normal and common operation that is prevalent among of a number of different applications. However, sync usually implies that the operations can go both ways. For example, when I sync my iTouch with my Gmail, any read message on Gmail becomes a read message on my iTouch and vice versa. In this case with the Class Scheduler, the operations are only one sided. This may present mild confusion.

Consider the following scenario: a student was creating his/her schedule but after some work, they realize that one class may conflict with another class. Some users may then feel inclined to drag classes between terms in order to promote ease of use.

User control and freedom: Does are system support a undo/redo system?

No undo/redo system is supported on this system. However, it would be understandable if the user would require certain aspects of this feature.

In terms of freedom, the user is allowed to schedule for any term at any point with a 4 year span and possibly beyond based on the following drawing that I observed.
However, there was a slight scenario that presented itself which still remains unanswered. What happens if there is a user who wishes to take a a class that is not within their major, but it still satisfies the requirement? An example, according to Tyler DeAdder, was taking ECE 353 in place of STATS 314. The point of bringing this up is that more freedom is required for EECS students who will be relying on this schedule completely when identifying classes that they should take.

Consistency and standards: Is the terminology that has been used consistent with the system and platform?

In terms of internal consistency, the interface is mostly sound. But when comparisons are drawn between the list of classes and the tree of classes, questions are raised. For example, a user expects the same linear listing method to be present throughout most of the interface, which includes both the initial page and the listing of individual classes that the student can take that term. Bring in a non-linear and non-expected structure to the user's interface and confusion will likely occur.

Error Prevention: Does the design prevent a problem from occurring? If an error occurs, does the the system provide a detailed enough message that related what went wrong with the program? If the user is able to use an action that may or may not cause an error, does the system ask for conformation before allowing the user to proceed?

The system attempts to prevent the user from making errors in a number of ways.

The first way is through removal of the functionality of hyper links and check boxes when a user has not completed a pre-requisite for a certain class. In addition, the user will be notified passively through the use of graphics (X's) and tool tips when the user hovers his or her mouse over the class that they wish to take.

In addition, if the user has a course that he/she will plan to take but did not satisfy the prerequisite requirement for that class, the system will highlight the description in yellow, allowing the user to proceed but passively warning them about the un-fulfilled requirements.

Recognition rather the Recall: Does the interface reduce the memory load of the user by making all of the objects, actions, and options visible when applicable?

The user interface looks more toward guided recognition. By showing the user what he/she can take in the coming terms through graphics and tool tips, the user is encouraged to register in a

guided fashion. However, it was unclear about whether or not the text would not appear, be grayed out, or have a red X if the user was not allowed to register for the class.

Flexibility and efficiency of use: Is it possible for the system to cater to both experienced and novice users through the speed at which the task can be completed? Can the user "tailor frequent actions"?

There are no shortcuts nor scripting allowed for this interface, thus removing the ability for the user to tailor the use frequent actions to complete tasks. However, the user is given the option to synch their schedule with their transcript. Also, the user can save their transcript, thus decreasing the time spent identifying classes to take in the next iteration of scheduling. Maybe the students could import the “suggested classes to take” rubric from advisors to assist in speeding up the process?

Aesthetic and minimalist design: Is all the information that is displayed intended to be there; does the displayed information on the GUI have relevant qualities?

All information present is predominately restricted to scheduling; aside from being able to browse classes and create schedules, the user will be able to update past terms with their grades and print out physical copies of their schedules. No additional features were given.