Darren Andreychuk

Overview

A Task-Centered Walkthrough is a low-cost way to evaluate, and debug, the interface of a system in the early stages of its development. It is the fourth phase in Task-Centered System Design which was introduced by Clayton Lewis and John Rieman in 1993. In the first phase of this methodology specific users of the system must be identified along with a set of real-world tasks that they complete using the system. These believable tasks must mirror what users actually do in their work environment but not how they do it. These tasks must be very specific and they must be complete. This information can be gathered by observing the actual target users working in their environment, interviewing someone who has a great deal of knowledge of the target user's needs, or if none of the users are available then the design team can come up with a list of target users and realistic tasks. During the second phase, the design team must decide which users and which tasks to include in the system. Next, each user and their corresponding task descriptions are merged with the system design. This combination is called a scenario and the goal is to tell a complete story.

With a manageable list of scenarios in place, a walkthrough of the system can be performed by the design team. A task-centered walkthrough can be performed by one or more people on the design team. In general, however, the more people that are involved in the walkthrough the better because people on the team can have differing perspectives thus helping the team identify problem areas in the system. The walkthrough may or may not include target users, but this can depend on how many walkthroughs have already been performed and if any of the users are available to perform the walkthrough.

How a Task-Centered Walkthrough Works

The main focus of a task-centered walkthrough, as well as a Pluralistic Walkthrough introduced by IBM in 1994, is to tell a story about a specific task that a particular user is trying to perform. These stories are very specific and are broken down into a sequence of steps. A scenario is chosen by the designer and a walkthrough is performed on it. To make the process of setting up and performing a task-centered walkthrough easier to understand, Lewis and Rieman proposed the following algorithm:

- Choose a task scenario from the list of scenarios
    - For each step in the task scenario do the following
        - Ask yourself: Can you build a believable story that will motivate the user's actions?
        - Ask yourself: Can you rely on the user's expected knowledge of the system to complete this step of the task?
        - If you cannot answer yes to either of these questions then

- You have located a problem in the interface
- Note the problem and any comments, or possible solutions, that come to mind

▪ Once a problem has been identified, assume that problem has been fixed

- Proceed to the next step in the task scenario

The system designer, or designers if many people will be carrying out the walkthrough, must place themselves in the mind of their target user and must stay on track with what that user is trying to accomplish during the scenario. Only when this mindset is established, and maintained throughout each scenario, can the above algorithm work effectively for the people performing the walkthrough. This algorithm is also effective when it is carried through to completion. Once a problem as been identified in the system, it should be noted and the walkthrough should continue as if it has been fixed. In other words, the walkthrough should be completed even if the system is so bad that it will need a complete redesign. This will help the design team generate new ideas and help them avoid similar mistakes in future designs.

When and Where a Task-Centered Walkthrough is performed

Task-centered walkthroughs are carried by the design team early in the design process. This means that the interface can be evaluated, and debugged, when little or none of the system has been implemented and it is merely laid out in a series of throw-away paper prototypes. When a prototype is created it is evaluated through a task-centered walkthrough. If revisions need to be made, the design team will make those revisions in newer prototypes. If revisions are not needed, however, the prototype can move on to the next stage where implementation will begin on it.

Benefits and Limitations of Task-Centered Walkthroughs

The main benefit of a task-centered walkthrough is that it is a low cost technique to locate problems early in a system's development. Since a task-centered walkthrough is carried out this early in the design process it can be performed on throw-away paper prototypes of the system. This is very effective because it allows the design team to catch smaller design issues before a significant amount of resources are put into the implementation of the system. These problems can be spotted by the design team as they move through the steps of a scenario and the necessary modifications can be made before the system moves on to the next phase of development. The drawback to this is that the walkthroughs are usually carried out in the same environment as the one the system is being developed in. While this adds to the low-cost argument of the walkthrough being performed in a lab instead of in the field, this may not be sufficient because the design team will not see the system being run in the real world.

Task-centered walkthroughs are also very flexible because they can be performed with or without the actual people who will be using the system. This means that the walkthrough can be carried out by system designers and developers on the project. However, as previously stated, system designers must keep themselves in the mindset of their target users when conducting this walkthrough. Another point to make here is that a user's time is very important to them and this must be respected. Typically, it is not a good idea to include users in early walkthroughs of the system because this is a time when smaller problems in the system are located and it is not necessary to waste their time with these problems. If actual users are to be included in future walkthroughs it is important to have trivial problems from previous walkthroughs resolved so that they do not appear when users are present.

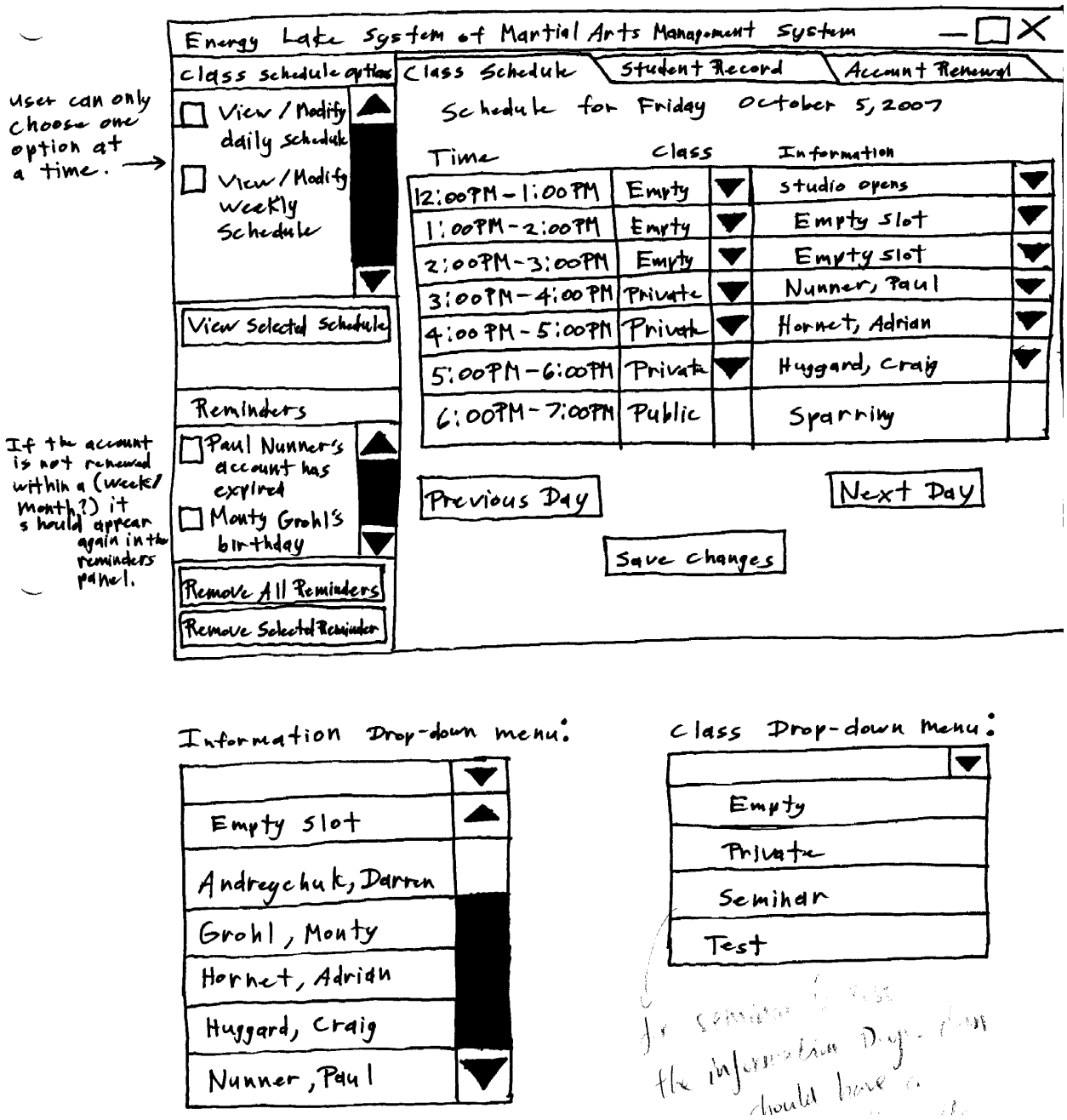
Another benefit of task-centered walkthroughs is that they can be performed by a single system designer, or a group of system designers as done by IBM in their pluralistic walkthrough. One point does need to be raised here, however. Even though a task-centered walkthrough can be performed by a single person it is better to do such a walkthrough with a group of people. In general, walkthroughs that are carried out by more than one person usually produce better results for the design team. For example, different people on the team, and even users if they are participating, can have different perspectives as to what caused a problem in the interface and how it can be fixed. However, since the walkthrough can be a group activity it must progress at the same pace for all group members. Sometimes the task scenario might halt at a single step because the participants are debating a problem in the interface. This can disrupt the natural flow of steps in the scenario and the people participating in the walkthrough will not gain a good understanding of how the steps are connected together.

Example

The Situation

The studio owner at the Energy Lake System of Martial Arts needs a software system that will assist him with the day-to-day affairs of his business. This system will be located on the only computer in the studio. Our target user is Gary Braniff. He owns, manages, and is the only instructor at Energy Lake. Gary is a typical computer user. He is able to use applications like Internet Explorer and Microsoft Word. Presently, Energy Lake has about a hundred students. Each day the studio is used by numerous students for private and public classes. Students can also use the equipment in the studio to workout. Currently, all student records are paper-based and are placed in manila folders. All student folders are kept in a filing cabinet. The date and time of each private class is kept in a notebook. Students can schedule, as well as cancel, private classes. The Energy Lake System of Martial Arts Management System will replace the old paper-based way of running the day-to-day affairs at Energy Lake. This system will also inform the instructor when a student's account has expired and needs to be renewed. This system will save the instructor money typically spent on student contract templates that are specially made for his business and day planners for the keeping track of the day-to-day class schedule.

Class Schedule Prototype



Example Task: Class Swap

Paul, a student at Energy Lake who will be testing for his blue belt this week, will not be able to make it to his private class on Friday at 3:00PM. The day before Paul calls Gary at the studio to tell him that he will have to cancel his class scheduled for tomorrow. Another student, Scott who is currently in the studio and is working out, over hears this conversation. Gary hangs up the phone and now needs to

remove Paul's name from the 3:00 time slot on Friday. Before he is able to do this, Scott walks up to Gary's desk and asks when Paul's class was supposed to be held. Gary tells Scott that it was supposed to be tomorrow at 3:00. Scott realizes that he has not been beaten up in over a week and wants to get some extra sparring practice. Scott quickly asks Gary if he can take Paul's time slot. Gary agrees to this and is delighted that the time slot will not be vacant. Scott thanks Gary for the time slot and walks over to the punching bags to continue his workout. Gary now wants to remove Paul's name and put Scott's name in the 3:00 time slot, but needs to do this as quickly as possible because he has another class starting in a few minutes.

Scenario 1: Last minute class cancelation

Step #	Description of the step	Does the user have enough knowledge or training to carry out this step?	Is the user motivated to do this step? Is it believable that they would do this?	Comments and suggestions
1	Gary answers his phone in the studio. It is Paul phoning to cancel his class tomorrow.	Yes	Yes	
2	Gary tells Paul that he will cancel the class.	Yes	Yes	
3	Gary sees Scott approach his desk and finds out that Scott wants to take Paul's class.	Yes	Yes	
4	Gary brings up the Energy Lake management program on his taskbar.	Yes	Yes	
5	Gary clicks on the schedule tab.	Yes	Yes	
6	Gary selects the View/Modify daily schedule in the Class	Yes	Yes	

	Schedule Options panel.			
7	Gary clicks the View Selected Schedule button.	Yes	No	After clicking on View/Modify daily schedule the daily schedule should appear. Gary should not have to click on a check box beside the option and then click on a button to access that option.
8	Gary finds Paul's time slot for that day in the Daily Schedule tab.	Yes	Yes, if the daily schedule is not very long.	Usually the studio is open for six to nine hours a day. This is minor problem since, on some days, the studio is open for a shorter length of time then on other days.
9	Gary moves his cursor down the Information column of the table and clicks on the drop down menu.	Yes	Yes, since there are only three columns in the schedule table.	The number of columns in the daily schedule table is likely to stay the same.
10	Gary scrolls down the list to find Scott's name.	Yes	No, he will have to scroll through a big list of students.	There are around a hundred students at Energy Lake and that number is growing. The student list will get longer. One possible solution is to add a student query. Gary can type the first letter of the student's last name and he would be taken to that area in the list.
11	Gary clicks on Scott's name.	Yes	Yes	
12	Gary views the updated time slot to see if it is correct.	Yes	Yes	
13	Gary clicks the save changes button.	Yes	Yes	
14	Gary minimizes the program on his taskbar.	Yes	Yes	

Bibliography

Greenberg, S. Working through Task-Centered System Design. In Diaper, D. and Stanton, N. (Eds) *The handbook of Task Analysis for Human-Computer Interaction.* Lawrence Erlbaum Associate, 2004. P 49-66.

This paper defines the four phases of Lewis and Rieman's Task-Centered System Design (1993) with a good example that demonstrates how to set up a task-centered walkthrough. This is a good paper to read before the task-centered walkthrough presentation. Focus on Phase IV: Evaluate via Task-Centered Walkthroughs, p 6-15. This paper can be found on the course webpage at:
http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2004-TaskAnalysis.LEAChapter.pdf

Lewis, C. and J. Rieman. Task-Centered User Interface Design: A Practical Introduction (1993).

This book is often referred to as the book that got Task-Centered System Design started. Focus on chapters two and four when researching Task-Centered Walkthrough. This book can be found on the course webpage at:
http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/HCIPapers/lewisriemanbook-index.html

Nielsen, J. and R. Mack. (eds) Usability Inspection Methods. Wiley and Sons, 1994. P25-62.

Chapter 3 of this book, p 63-76, provides a detailed description of the pluralistic usability walkthrough developed by IBM. This chapter also provides a good list of benefits and limitations the pluralistic usability walkthrough has. This chapter is on the course webpage at:
http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/HCIPapers/nielsen94-inspection-ch3-pluralistic.pdf

Greenberg, S. Overview of Task Centered Walkthrough, CPSC 481 Slide deck.

http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/CPSC681/TaskCentred.pdf