

ProfCast

Heuristic Evaluation

Team Marvel

Mail Box 265

Chris Demeniuk

Yarun Luon

Josh Morse

Melissa Perez

John Suciu

SI 622: Evaluation and Systems and Services

Assignment 6

Table of Contents

Table of Contents	2
Table of Figures	3
Product Description	4
What is ProfCast?	4
How Does it Work?	4
Target Population for this Product.....	4
Overview of this Report.....	5
Scenario One – Recording a Lecture	6
Overview	6
The Main Recorder Window	7
Recording in a PowerPoint slide.....	10
The Publish and Pause Recording Dialog.....	12
The Publishing Toolbar.....	15
Scenario Two – Managing ProfCast Settings	17
Overview.....	17
Podcast Manager.....	17
Podcast Series Dialog and Help.....	23
Settings Dialog.....	28
Summary.....	31
Usability.....	31
Functionality	32
Aesthetics	32
Conclusion	32
References.....	33

Table of Figures

Figure 1 - Main Recorder Window	7
Figure 2 – Inadequate Feedback	8
Figure 3 - Recording Button Grayed Out	9
Figure 4 - Recording and Published Grayed Out	10
Figure 5 - The Publish and Pause Recording Dialogue	12
Figure 6 - Two Interfaces with Similar Options	13
Figure 7 – The Resume Recording Dialogue	13
Figure 8 - Publish Button	14
Figure 9 - The Publishing Toolbar	15
Figure 10 - Save in the Publishing Toolbar	15
Figure 11 - Three Verbs in the Publishing Toolbar	16
Figure 12 – The Enhanced Podcast Dialogue	16
Figure 13 - Podcast Manager	18
Figure 14 - Gear Button	19
Figure 15 - Series vs. Episode.....	20
Figure 16 - Golden Orb Icon.....	21
Figure 17 - Explicit Dropdown List.....	22
Figure 18 - Podcast Series dialog.....	23
Figure 19 – The Podcast Series Dialogue	24
Figure 20 - Help Icon	25
Figure 21 - Navigating through Help.....	26
Figure 22 - Help Overlay	27
Figure 23 - Settings Dialog	28
Figure 24 - Getting to the Episode Settings Screen: Step 1	29
Figure 25 - Getting to the Episode Settings Screen: Step 2.....	29
Figure 26 - Getting to the Episode Settings Screen: Step 3.....	30

Product Description

What is ProfCast?

Lecturers teaching a class to students using Microsoft® PowerPoint or Apple® Keynote often wish that they could make their lecture available online. Unfortunately, they are often left with only the option of providing the slides for students to read. The problem with this is that students reviewing the slides often miss out on additional information provided by the lecturer. ProfCast allows a lecturer to capture this information by creating a 'podcast' of the slides, synced with audio from the presentation, for the students to download and view later. Students can then conveniently review both the visual slides and the auditory components of the lecture together.

How Does it Work?

As mentioned above, the user must create a presentation in either PowerPoint or Keynote. The user then launches the ProfCast application. In the Main Recorder Window, the presentation files are then added by dragging them into the window or browsing for them. After this is completed, the next step is to select the audio input device in the drop down menu of ProfCast. One can select the default option (using the audio device of the computer), or select a different option (such as an external microphone) to increase the quality of audio input.

The third step is to click the *Start Recording* button, which will start the presentation from the first slide in PowerPoint or Keynote. The presenter begins speaking as if they were giving the presentation to their students, progressing through the slides until the presentation is finished. If the presenter needs to pause the recording, they simply press the escape button. This will display a pop up window with the option to either continue recording or publish the presentation. After the presentation is finished, the same window appears, and users should select *Publish*.

Publishing in ProfCast opens a window with several options. A user can encode the podcast through ProfCast's built-in Podcast Manager, then upload or email it. Similarly, the 'Save' option allows a user to save the recording quickly and create a podcast for upload later. The 'GarageBand' option helps modify the audio segment of the presentation (e.g. if the user needs to delete or change audio sections of their presentation, etc). The 'Send to iWeb' option creates an enhanced podcast to be uploaded and used in iWeb. The final result of any method is a podcast that can be viewed on iTunes or on an iPod.

Target Population for this Product

The target population for this product is instructors who have prepared Keynote or PowerPoint presentations for their lectures. The term "instructors" includes professors and teachers, but through the interview process we have learned that this group also

includes members of administration wishing to preserve the presentations of visiting guests, graduate student instructors of discussion sections, and information technology support staff wanting to service all of the groups above. We initially conducted our evaluation with these different groups in mind. However we were pleasantly surprised to learn that ProfCast served an even broader range of users than our interview results first led us to believe. We will touch upon some of these groups later in this report.

Overview of this Report

Our team conducted a heuristic evaluation of ProfCast. We made use of the ten heuristics developed by Jakob Nielson, which are:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation (Nielson, 2005)

Our team of five people tested these heuristics in two rounds. In the first round, each member took two heuristics. Each member went through the scenario, keeping the user's needs and their particular heuristics in mind as they used ProfCast. Each member noted any problems and what heuristic those problems violated. After the first round, each member took two different heuristics and tested another scenario, again taking notes about what heuristics were violated and how they were violated.

Each of our scenarios was designed to test certain interfaces in ProfCast. As a result, we decided to organize our results by scenario, and by the interface that was tested. This method would make it easier for developers to focus on problems within specific interfaces that they are working on. The problems we found are listed for each interface, along with the heuristic and how it was violated. We also suggested solutions that we had thought of, though we didn't focus on finding a solution to each problem at this point.

Scenario One – Recording a Lecture

Overview

We used the scenario that we created for Amanda Waller, our low-tech professor, for the first scenario to evaluate. The steps are listed below. We expanded upon these steps when the heuristic called for it. For example, we might have opened the help documentation to see how easy it was for the user to figure out how to accomplish a specific step.

1. The user launches ProfCast on Mac OSX and is greeted by the Main Recorder window.
2. The user checks to see that the audio input level is correct.
3. The user loads two PowerPoint slides, one by dragging it into the window and one by clicking the plus button.
4. The user presses the start recording button to open PowerPoint and begin recording.
5. The user goes through their PowerPoint slides, pressing escape after several slides to pause the recording.
6. The user resumes their recording and progresses to the end of their slides.
7. At the end of the presentation, the user selects the “Publish” option on the dialog box.
8. On the next dialog, the user selects the save option and saves the podcast.
9. The user then presses the “Export enhanced podcast” button and saves the lecture as an advanced podcast.

This scenario tests the following interfaces:

- The Main Recorder Window
- ProfCast information during the PowerPoint presentation
- The pause/publish recording dialog
- The post-recording dialog with editing and exporting options
- The save dialog
- The export dialog

The Main Recorder Window

The Main Recorder Window appears when ProfCast is launched. Its main purpose is to facilitate recording by loading presentations and launching them while recording audio (See Figure 1).



Figure 1 - Main Recorder Window

The following issues were discovered while evaluating the Main Recorder Window:

Issue	Problem	Heuristic
1	User does not receive adequate feedback about why the file was not added to the Main Recorder Window	Help users recognize, diagnose, and recover from errors

Description

If an inappropriate object is placed in the application (e.g. not a presentation file), the file just quickly moves back to where it came from. There is no dialog saying what is wrong, and the user is unable to achieve their goal, and may not know why. This violates the heuristic that systems should 'help users recognize, diagnose, and recover from errors', because the user does not receive adequate feedback about why the file was not added to the Main Recorder Window (See Figure 2).



Figure 2 – Inadequate Feedback

Issue	Problem	Heuristic
2	User doesn't have any indication of why they can no longer record or add new presentations	Visibility of system status

Description

If the user tries closes the Podcast manager and goes back to the Main Recorder Window, the 'Resume Recording' button is dimmed (inoperable). Dragging a new presentation file into the window produces no change in state (See Figure 3). This violates the heuristic of 'visibility of system status,' because the user doesn't have any indication of why they can no longer record or add new presentations.



Figure 3 - Recording Button Grayed Out

Issue	Problem	Heuristic
3	Main Recorder Window is used to publish in some situations and record in others	Consistency and standards

Description

Similar to the problem above, either the Publishing and Recording button are always dimmed on the Main Recorder Window, depending on whether the user just started ProfCast (Publish is dimmed) or came back from the Podcast Manager (Record is dimmed). This is a violation of the heuristic of 'consistency and standards', because the Main Recorder Window is used to publish in some situations and record in others, but nothing alerts the user to why their options change for the same User Interface (See Figure 4).



OR



Figure 4 - Recording and Published Grayed Out

Recording in a PowerPoint slide

While not technically a ProfCast interface, ProfCast does launch the PowerPoint application and records from the background. The lack of an interface proved to be a major heuristic violation. Problems are listed below:

Issue	Problem	Heuristic
4	User is not informed that the system is recording in the background	Visibility of system status

Description

There is no feedback that ProfCast is recording while the PowerPoint presentation is running. This violates the ‘visibility of system status’ heuristic because the user is not kept informed that the system is working in the background once they press the recording button.

Issue	Problem	Heuristic
5	User does not know that ProfCast is not recording if they launch their presentation outside of ProfCast	Help users recognize, diagnose, and recover from errors

Description

Similarly, the user does not know that ProfCast is not recording if they launch their presentation from PowerPoint rather than starting it with the ‘Start Recording’ button. This violates the heuristic ‘help users recognize, diagnose, and recover from errors’ because in this instance the user is likely making an error, and will likely not learn that the lecture failed to record until after it is over, which could prove to be very frustrating.

Issue	Problem	Heuristic
6	The user does not know whether pressing a hotkey will start or stop recording	Visibility of system status

Description

There is no feedback if the user presses the ‘stop/start recording’ hotkey. The presentation will continue, but the user has no way to know if recording is taking place. If they accidentally press the key too many times, they may be not recording when they think they are recording. This violates the ‘visibility of system status’

heuristic because the user does not know whether pressing the hotkey again will start or stop recording. This could be remedied by displaying recording status, and also by creating separate hotkeys to start and stop recording.

Issue	Problem	Heuristic
7	Hotkeys used to start and stop recording are not shown to the user	Recognition rather than recall' heuristic

Description

The hotkeys used to start and stop recording are not shown to the user at any time before or during a presentation. This forces the user to remember these very important hotkeys from the settings dialog, which is a violation of the 'recognition rather than recall' heuristic.

Issue	Problem	Heuristic
8	User cannot see recording levels while recording	Visibility of system status Help users recognize, diagnose, and recover from errors

Description

The user cannot see audio levels from the microphone while recording. This violates the 'visibility of system status' heuristic, because while the user can see microphone levels before recording, they have no way of knowing if the levels are correct once they begin recording. It also violates the heuristic 'help users recognize, diagnose, and recover from errors,' because if the microphone level suddenly drops, the user has no way of knowing about this problem until after the recording is over.

Issue	Problem	Heuristic
9	No visible exit from a presentation while recording	User control and freedom

Description

All ProfCast controls are hidden while recording. This is a violation of the 'user control and freedom' heuristic, because the user has no visible exit from a presentation or access to ProfCast controls until the presentation is finished, which limits the user's control of the system.

The Publish and Pause Recording Dialog

The dialog depicted in Figure 5 appears either when a PowerPoint presentation is finished or when the presentation is stopped early by pressing the escape key or right clicking and selecting 'End Show' from within a presentation. We found several issues with this dialog in our evaluation.

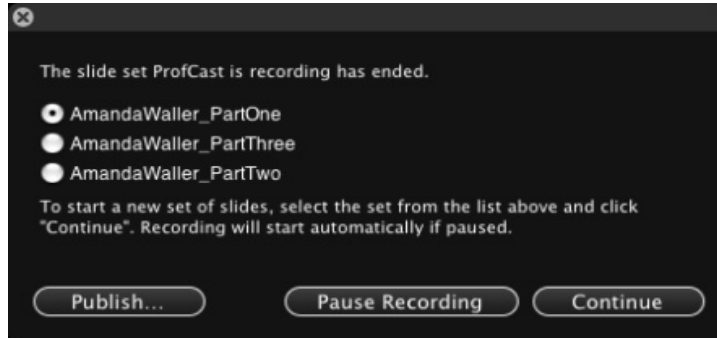


Figure 5 - The Publish and Pause Recording Dialogue

Issue	Problem	Heuristic
10	Two interfaces on the screen that have similar options	Aesthetic and minimalist design

Description

When a user exits the presentation, they see the dialog in Figure 5, however the Main Recorder Window is still visible, and the 'Start Recording' button changes to a 'Pause Recording' option, while the 'Publish' button becomes available (See Figure 6). This violates the heuristic of 'aesthetic and minimalist design,' because there are two interfaces on the screen that have similar options, which could confuse the user.

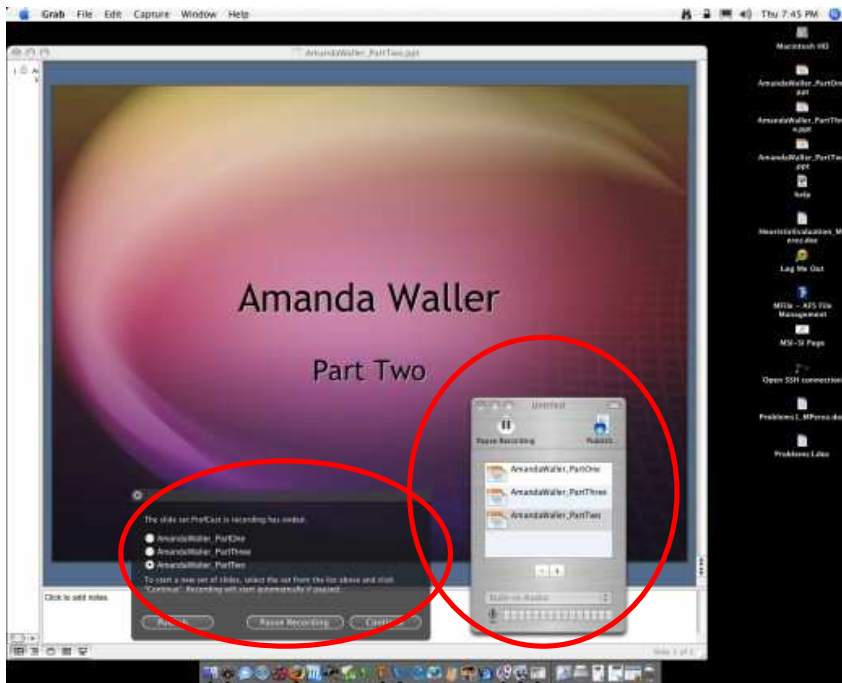


Figure 6 - Two Interfaces with Similar Options

Issue	Problem	Heuristic
11	The user presses a button and sees no change in state	Visibility of system status

Description

Pressing the ‘Pause Recording’ Button changes its text label to ‘Resume Recording,’ but otherwise has no visible effect, because to get this dialog the user must have exited the presentation, and hence paused recording (See Figure 7). This violates the ‘visibility of system status’ heuristic, because the user presses a button and sees no change in state, which could confuse them.

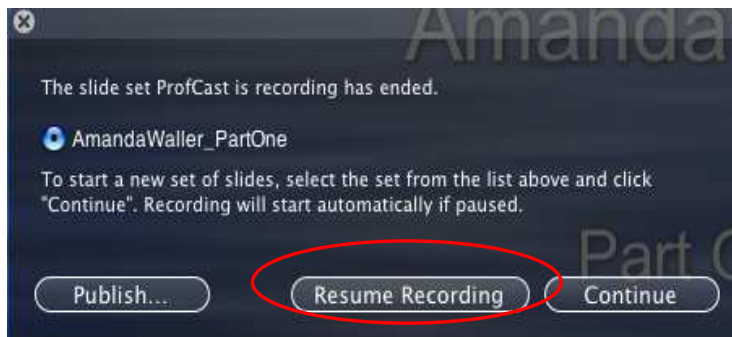


Figure 7 – The Resume Recording Dialogue

Issue	Problem	Heuristic
12	Publishing button does not appear to logically relate to the recording window	Match between system and real world

Description

The Publish button is used to actually stop recording and move to the next step in the process. This violates the ‘match between system and real world’ heuristic because the publishing button does not appear to logically relate to the recording window, and there is no ‘End’ or ‘Stop Recording’ button. The closest is ‘Pause Recording’, but unfortunately, as outlined above, this button has its own problems, and thus the user could become very confused about how to stop the recording, when the only necessary action is press the button labeled “Publish” (See Figure 8).

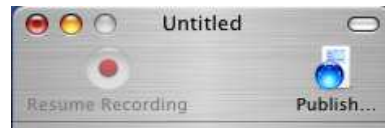


Figure 8 - Publish Button

Issue	Problem	Heuristic
13	User expects that the word “Continue” means that the presentation will begin from where they exited	Consistency and standards

Description

When the user presses the ‘Continue’ button after exiting in the middle of a presentation, the presentation will start over from the beginning. This violates the heuristic ‘consistency and standards,’ because the user expects that the word “Continue” means that the presentation will begin from where they exited. Better phrasing might be something like ‘Restart Presentation.’

The Publishing Toolbar

This toolbar appears on the screen when the user presses the 'Publish' button, either in the 'Publish and Pause Recording' Dialog or in the Main Recorder Window (See Figure 9).

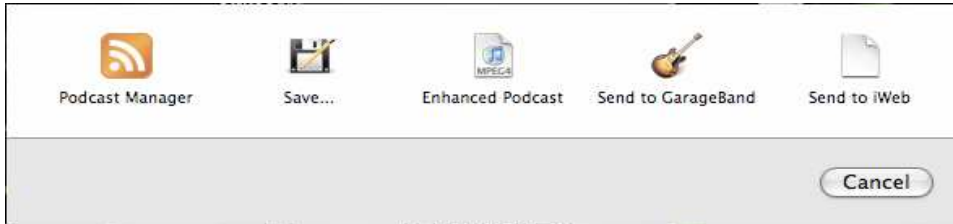


Figure 9 - The Publishing Toolbar

Issue	Problem	Heuristic
14	When user accidentally cancels their save, there is no easy way to get the option to save again	help users recognize, diagnose, and recover from errors

Description

If the user chooses 'Save', then presses 'Cancel' in the 'save file name' dialog box, the Publishing toolbar disappears. The user can open it again from the (still visible) Main Recorder window, but that isn't clear. This is a violation of the 'help users recognize, diagnose, and recover from errors' heuristic because if the user accidentally cancels their save, there is no easy way to get the option to save again, and the user could think their presentation has been lost (See Figure 10).



Figure 10 - Save in the Publishing Toolbar

Issue	Problem	Heuristic
15	Inconsistent terminology in the Publishing window	Consistency and standards.

Description

The terminology of the five options in the publication window is inconsistent and not very descriptive. Three of the five choices use verbs (Save, Send to GarageBand, and Send to iWeb) while the other two use nouns. This can cause problems for the users because without the verb they don't know what 'Podcast Manager' and 'Enhanced Podcast' actually do. The verb "Send" is not descriptive and also doesn't tell the user much about those options. This is a violation of the heuristic 'consistency and standards.' We would suggest the following terminology: "Manage podcast in Podcast Manger", "Save as ProfCast recording", "Save as Enhanced Podcast", "Edit in GarageBand", "Upload to iWeb" (See Figure 11).



Figure 11 - Three Verbs in the Publishing Toolbar

Issue	Problem	Heuristic
16	User cannot distinguish between two options	Match between system and the real world

Description

The nature of Enhanced Podcast is not entirely clear to users who do not know what an enhanced podcast is, and could easily be confused with the 'Save' option (See Figure 12). This is a violation of the heuristic 'match between system and the real world,' because the system is using confusing terminology. Both options save something. Unfortunately the format for what gets saved is different depending on which option is chosen. Using the wording described above might help to make that clearer to the user.



Figure 12 – The Enhanced Podcast Dialogue

Scenario Two – Managing ProfCast Settings

Overview

Our second scenario was designed to allow us to test interfaces that we had not tested in our first scenario: in particular, the Podcast Manager and the ‘Settings’ dialog box. We assumed the user had already recorded their lecture, to avoid overlap with our previous scenario. The steps we took are as follows:

1. The user begins from the ‘Publishing’ toolbar after making a recording
2. The user opens the Podcast Manager from the toolbar
3. The user creates a new series and episode in the Podcast Manager and sets up details for them.
4. The user presses the upload button to upload via FTP, and is informed they need to set up a server.
5. The user configures an FTP server in the ‘Settings’ dialog.
6. After the server is configured, the user configures episode setting so that they have mono sound, and half-size slides.
7. The user checks to make sure their version of ProfCast is up-to-date with the ‘Software Update’ tab
8. On the ‘General’ tab, the user decides to change the screen capture hotkey to F1.
9. The user then uploads the podcast to the FTP server on the ‘Server’ tab of the ‘Settings’ dialog.

Podcast Manager

The Podcast Manager is most frequently launched from the ‘Publishing’ toolbar, though it can also be launched from the ProfCast ‘File’ menu (See Figure 13). The Podcast manager is used to manage podcasts, particularly those that are made up of multiple recordings (episodes). It is a midway point for the user to configure podcasts and enter information about them before uploading them to a server. We discovered a number of heuristic violations through our testing of this interface.

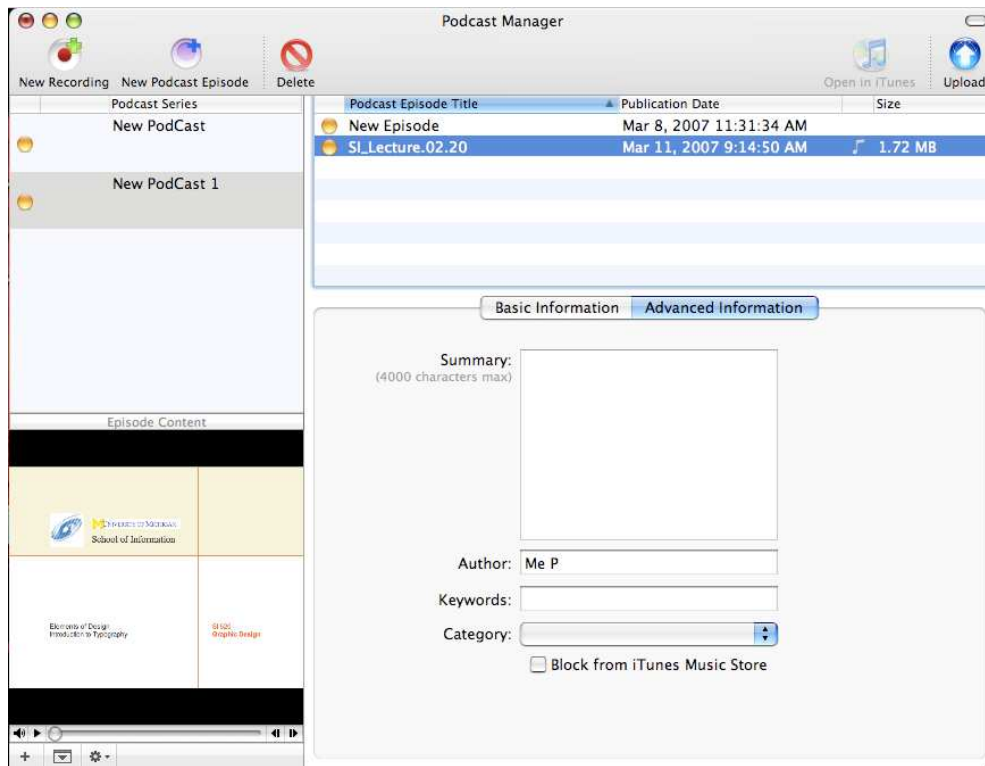



Figure 13 - Podcast Manager

Issue	Problem	Heuristic
17	Icon does not relate to its intentions	Match between system and real world

Description

Clicking on the gear button () in the lower left side of the Podcast Manager brings up another menu with options to add new recordings and add a new podcast series (See Figure 14). This violates the heuristic ‘match between system and real world’ because a user would likely not associate a gear with options for managing a podcast series, since gears are normally used in machinery. Even in ‘accepted’ computer analogy, a gear is often used for system or device configurations.

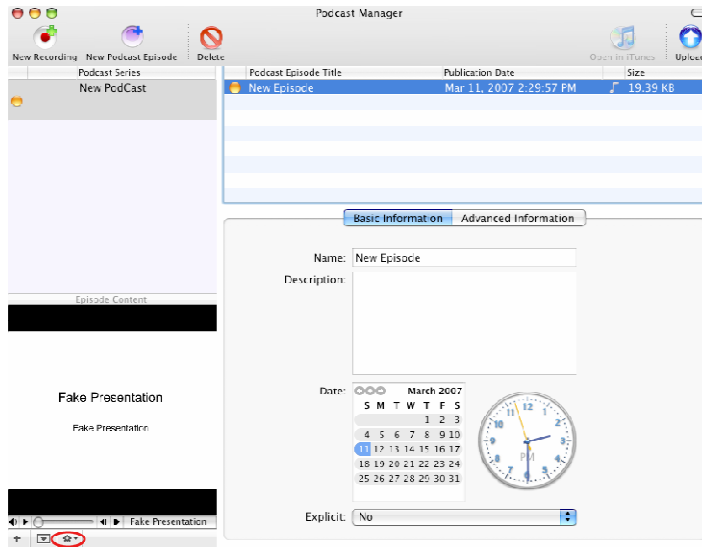


Figure 14 - Gear Button

Issue	Problem	Heuristic
18	User cannot easily move between interfaces and steps	User control and freedom

Description

From the Podcast Manager, there is no clear way to get back to the Publishing toolbar. If the Podcast Manager is closed, the user is brought back to the Main Recorder window, which still has no obvious way to get back to either the Publishing toolbar or the Podcast Manager. This violates the heuristic of 'user control and freedom,' because the user cannot easily move between interfaces and steps. This also violates the heuristic of 'help users recognize, diagnose, and recover from errors,' because if the user accidentally closes the Podcast Manager, there is no clear way to get back to it.

Issue	Problem	Heuristic
19	User does not know the difference between series and episodes	Recognition rather than recall

Description

The Podcast Manager allows the user to manage two major data objects: Podcast Series and Podcast Episodes. The left side of the Podcast Manager is mostly devoted to managing Podcast series, and the right side to Podcast episodes (See Figure 15). A Podcast Series has one or more Podcast Episodes associated with it. Unfortunately, the hierarchical relationship between these two objects is not clear from the layout. This is a violation of the ‘recognition rather than recall’ heuristic, because the system does not allow the user to easily recognize the pattern between series and episodes. Instead, the user has to recall what each object is every time they open the Podcast Manager.

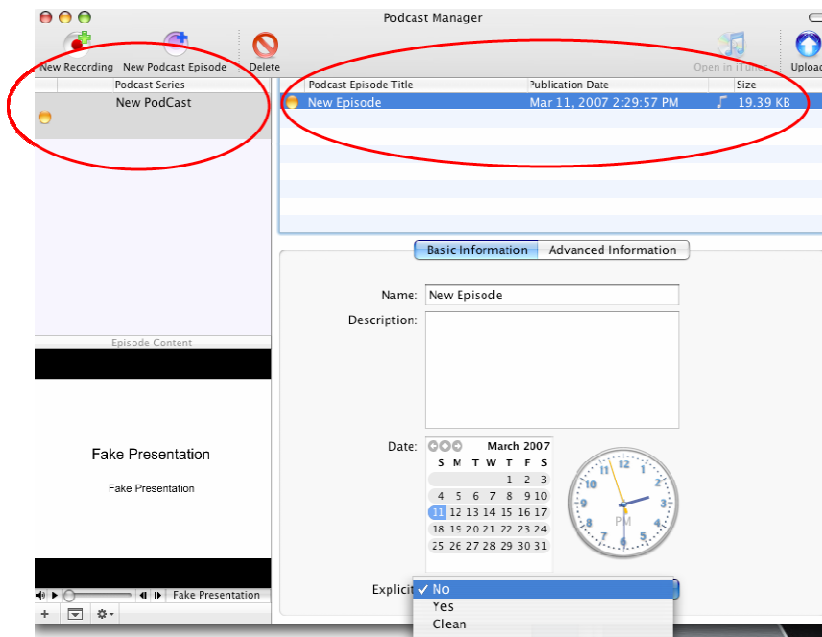



Figure 15 - Series vs. Episode

Issue	Problem	Heuristic
-------	---------	-----------

20	Using existing symbolism for a new meaning	Match between the system and the real world
----	--	---

Description

A golden orb icon () appears next to each Podcast Episode and Podcast Series. This is a violation of the ‘match between the system and the real world’ heuristic, because a user often associates a yellow circle with a traffic-light-style warning, but in this instance, the yellow circle is simply used as a bullet point (See Figure 16).

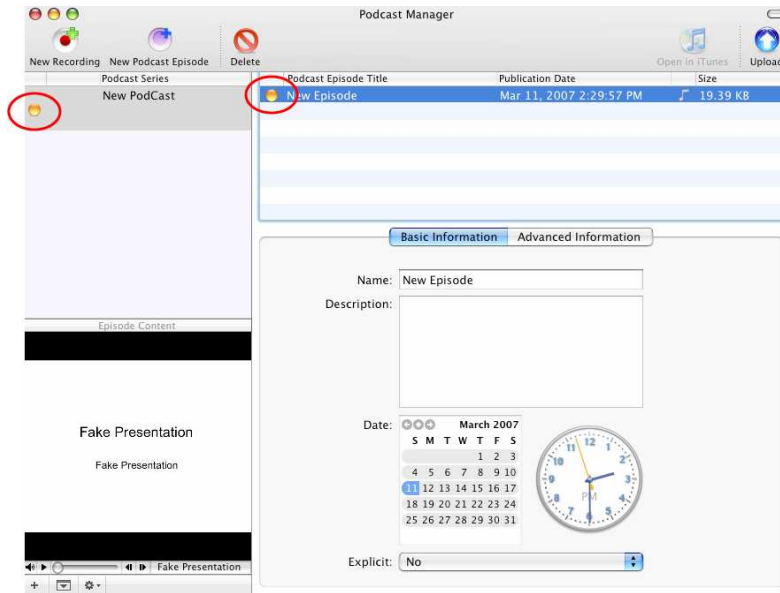


Figure 16 - Golden Orb Icon

Issue	Problem	Heuristic
21	Technical terms may confuse the user	Match between the system and the real world

Description

In the ‘Basic Information’ option for Podcast Series, there is a drop-down menu labeled “Explicit” with the options of “Yes”, “No”, and “Clean” (See Figure 17). This is a violation of the heuristic ‘match between the system and the real world,’ because the description uses technical terms that may confuse the user, who likely will not understand what “Explicit” means.

ProfCast Survey Results

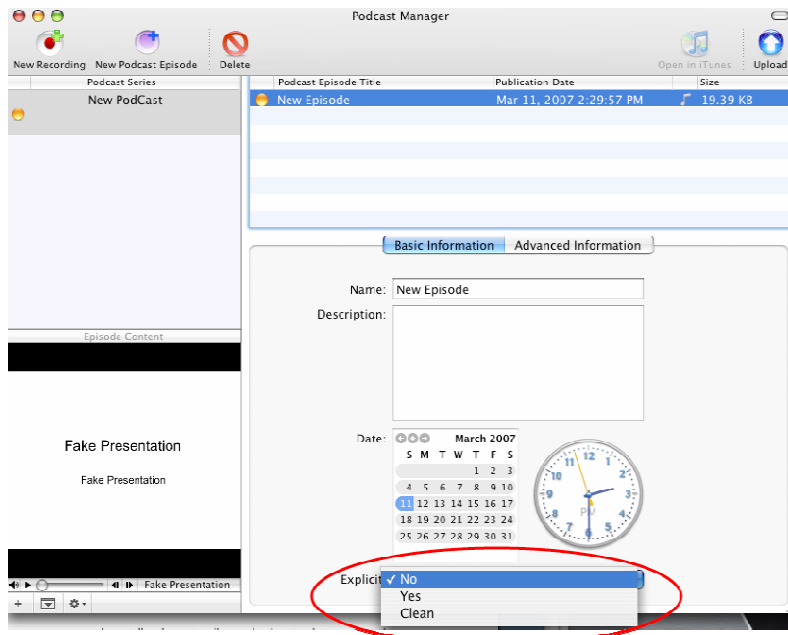


Figure 17 - Explicit Dropdown List

Podcast Series Dialog and Help

When the user chooses to create a new Podcast Series, a dialog window appears on top of the Podcast Manager. It cannot be moved, and there is no label to indicate that it contains information about a new Podcast series. This dialog is also appears when the user tries to upload a Podcast Series. We found a significant problem with this dialog, which will be discussed below. We also found problems with using the help documentation, which can be linked from this dialog.

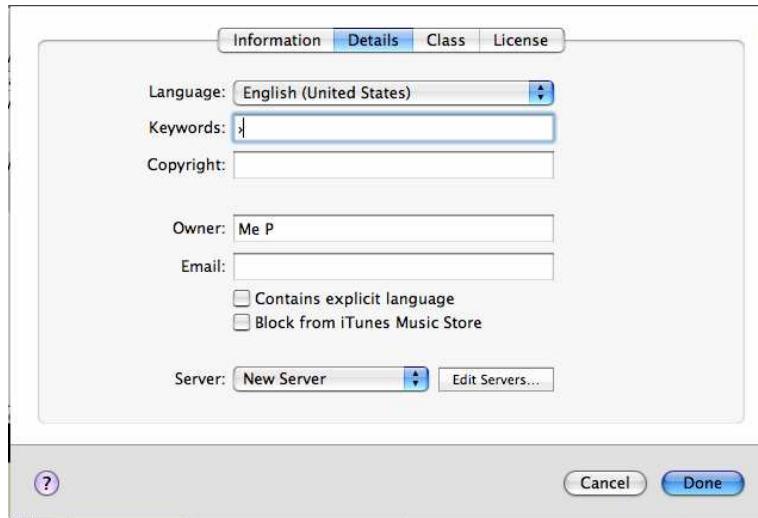


Figure 18 - Podcast Series dialog

Issue	Problem	Heuristic
22	Same dialog is used for more than one task	Consistency and standards Visibility of system status

Description

The 'Podcast Series' dialog does not have a title to indicate its purpose (See Figure 19). This is a violation of the 'consistency and standards' and 'visibility of system status' heuristics, because the same dialog is used for more than one task (managing Podcast Series information, creation of a new Podcast Series, and uploading a Podcast Series) but is not titled with the type of activity the user is doing. This could be remedied by labeling the dialog with the title "Create New Podcast Series" or "Manage Podcast Series Information", depending on what task the user is doing.



Figure 19 – The Podcast Series Dialogue

Issue	Problem	Heuristic
22	Unintuitive labeling of settings	Recognition rather than recall

Description

Server editing is done in the ‘Details’ tab of the Podcast Series Dialog. This dialog opens to the Details tab when the user chooses the ‘Upload’ option. This violates the heuristic of ‘recognition rather than recall’ because ‘Details’ is not intuitive as a location for configuration settings. One has to recall that server settings are located here.

Issue	Problem	Heuristic
23	User does not have access to the same icon in all interfaces	Consistency and standards

Description

The help icon appears on some dialogs, such as the ‘New Podcast Series’ dialog, but not others, such as the ‘Publishing’ toolbar and Podcast Manager. This is a violation of the ‘consistency and standards’ heuristic, because the user does not have access to the same help icon (?) from all interfaces.

This also violates the ‘help and documentation’ heuristic, because it is not easy for the user to get the help they need if the button doesn’t appear (See Figure 20). They will have to go to the help menu and search for the topic they need, rather than immediately being linked to the relevant help topic, as this button does.

Comment [d1]: Confused by what this means.

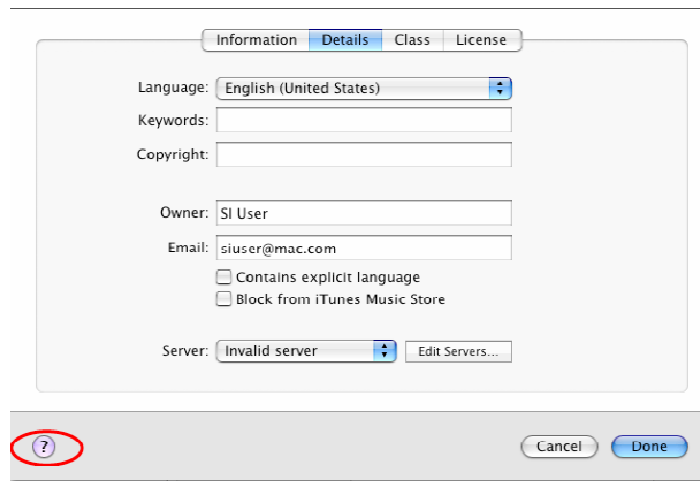


Figure 20 - Help Icon

Issue	Problem	Heuristic
24	Help documentation is not simple to search	Help and documentation

Description

When navigating through different help menu screens, there is no “history navigation” (See Figure 21). This violates the ‘help and documentation’ heuristic, because Help should be simple to search. When a user navigates through several menus in the documentation (via hyperlinks), there is no navigation that allows them to see the path they have taken to arrive at the current window. The user may want to jump back several help pages, rather than jumping back one page at a time.

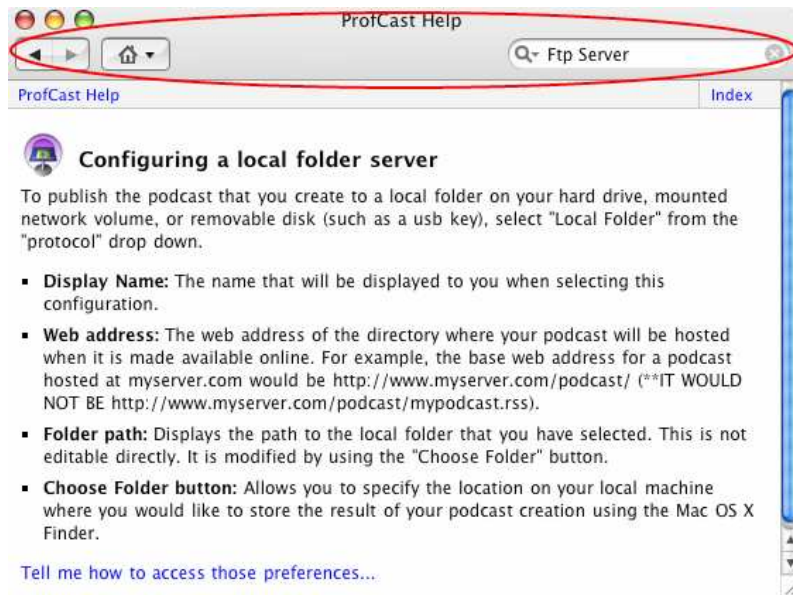


Figure 21 - Navigating through Help

Issue	Problem	Heuristic
25	Help documentation takes up entire screen	Help and documentation

Description

When the user opens the Help, it is overlaid on top of the ProfCast window and Podcast Manager (See Figure 22). This violates the heuristic of ‘help and documentation,’ because when the user follows a step-by-step guide, they need to execute actions as they read the help documentation. Having the menu covering the ProfCast windows forces the user to memorize information.

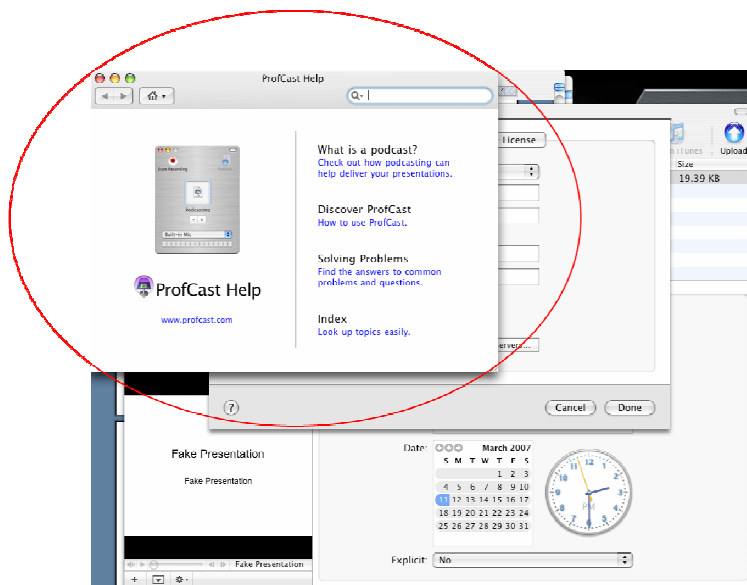


Figure 22 - Help Overlay

Settings Dialog

The Settings Dialog Window contains five tabs that manage most of the background settings for ProfCast (See Figure 23). This window appears when the user tries to edit settings, either through the multiple-step menu options, or by clicking the 'Upload' button in the Podcast Manager, which takes the user to the 'Servers' tab. If a user's trial period expires, they will be taken to this window with the Registration tab selected.

Comment [d2]: Does this violate the impartiality that we are supposed to be exhibiting?



Figure 23 - Settings Dialog

Issue	Problem	Heuristic
26	User must recall a long series of menu tasks rather than recognizing a simple button	Recognition rather than recall

Description

Getting access to General/Servers/Software Update/Registration/Episode Settings window is not intuitive. For example, if the user wanted to access episode settings they would have to click 'Edit Server' in the Podcast Series Dialog or to choose 'Edit Podcast Series' > 'Details' > 'Edit Servers' > 'Episode Settings' (See Figure 24-26). This violates the heuristic of 'recognition rather than recall' because the user must recall a long series of menu tasks rather than recognizing a simple button or menu option in the Podcast Manager.



Figure 24 - Getting to the Episode Settings Screen: Step 1

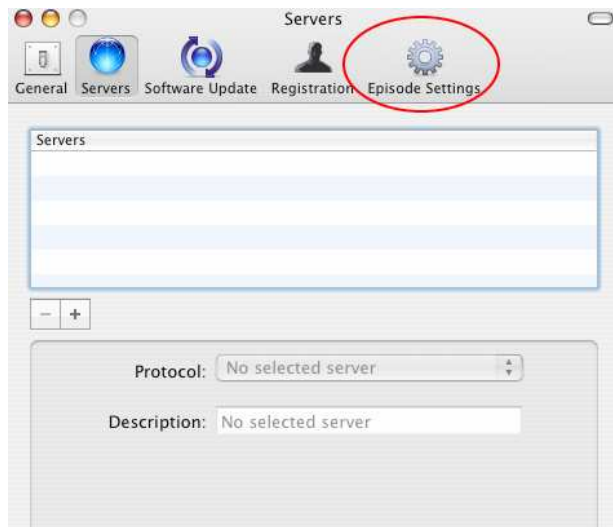


Figure 25 - Getting to the Episode Settings Screen: Step 2



Figure 26 - Getting to the Episode Settings Screen: Step 3

Issue	Problem	Heuristic
27	No shortcut keys for common configuration actions	Flexibility and efficiency of use

Description

There are no shortcut keys for common configuration actions, such as moving between tabs. This violates the ‘flexibility and efficiency of use’ heuristic by not allowing keyboard/key combination shortcuts, which would cut down on time for experienced users, who are the users most likely to use the Settings Dialog.

Issue	Problem	Heuristic
28	System does not check for well-formed ftp addresses	Error prevention

Description

If the user types an obviously false FTP address in the server tab (such as “This is not going to work.com.com.com”) the system still tries to connect, and will only give an error message if the connection fails. This is a violation of the ‘error prevention’ heuristic, because the system should be able to parse and check for obvious errors in address syntax and form, and not force the user to wait for a connection timeout.

Summary

Our team conducted a thorough heuristic evaluation of ProfCast using Jakob Nielson's ten heuristics. Each of our five team members tested two heuristics on our first scenario, which focused on the interfaces for recording and saving lectures. Each member then tested two different heuristics while running through our second scenario, which focused on managing and publishing Podcasts, and configuring ProfCast settings. Our major findings are listed below in the categories of usability, functionality, and aesthetics. We go into further detail about each issue we found in the following section.

Usability

There are many positive things that can be said about the usability of ProfCast. The WIMP (Windows, Icons, Menus, and Pointers) interface of ProfCast does a good job of supporting 'recognition rather than recall', with new buttons and dialogs appearing with relevant options and the user navigates through the process of recording, editing, and sharing a presentation. 'Error prevention' is generally good, as users are notified of common mistakes they might make. For example, if the user tries to start a recording without adding a PowerPoint or Keynote file, a message appears to notify them that they will only be recording audio.

While there are many good things to be said about the usability of ProfCast, most of the issues we uncovered dealt with usability. The most severe problem involved the total lack of feedback about ProfCast during a PowerPoint presentation. Users could easily finish a lecture before realizing that ProfCast had not been recording it correctly, and by then it would be too late. This is a major violation of the heuristic 'visibility of system status'.

There were a number of problems with the labeling of buttons and forms in ProfCast. The options "Save" and "Enhanced Podcast" in the Publishing Toolbar are good examples. Both options save the recorded lecture. However the different options save it in different file formats. From the labels, however, it isn't clear that 'Save' saves a ProfCast file, and "Enhanced Podcast" actually saves an enhanced podcast. By simply using a verb and a noun for both labels, this problem could be alleviated, but as it is there is a violation of the heuristic of 'consistency and standards'. Other examples of poor labeling are noted in the relevant interface sections.

ProfCast also suffers from a design that forces the user to follow a specific sequence of recording, managing, and uploading podcasts. Unfortunately, it is very difficult to navigate to a dialog or window that isn't next in your sequence. The Publishing Toolbar, which provides a central point from which a user can save, edit, upload, and manage podcasts, disappears after a user selects an option, and it isn't particularly intuitive for a user to get back to. For example, if a user opens the Podcast manager and then decides they want to edit the podcast in GarageBand, there is no obvious way to get to that option. This violates 'recognition rather than recall' because the user has to remember how to get back to the dialog through the menu system. It also violates user control and

freedom, because the user has to fight to do the steps of creating a podcast in a different order.

We found some problems with help documentation as well. While some interfaces have a useful help icon that loads the relevant help page, many do not. This is a clear violation of 'consistency and standards', as the user may come to expect that help documentation will be quickly accessible from all windows and dialogs. We also found problems with the help documentation itself, particularly in the way it overlays the actual application.

Functionality

Our heuristic testing did not uncover any major problems with the general functionality of ProfCast. The ability to record a lecture, save it, and publish it on the web generally works well once a user can overcome the usability problems associated with these tasks.

We did discover a major gap in the functionality of ProfCast while recording. ProfCast lacks any sort of visual feedback while a user is presenting PowerPoint slides. As mentioned previously, this lack of functionality causes serious usability issues, and needs to be addressed with a new feature that shows recording feedback during a presentation.

Aesthetics

ProfCast generally performs strongly in the area of aesthetics. We found that the graphical representations used by ProfCast, with few exceptions (like the gear icon in the Podcast Manager) map well to what a user would expect, and thus satisfy the heuristic of 'recognition is better than recall'. Examples include the New Recording button, the buttons on the Publishing Toolbar, and the clock and scheduler graphics in the Podcast Manager.

The overall layout of the dialogs and windows in Podcast is fairly clean, with minimal unnecessary information. This is particularly true of the most important interfaces, including the Main Recorder Window, the Publishing Toolbar, and the Podcast Manager. The Settings Dialog is a little cluttered, but overall ProfCast satisfies the 'aesthetics and minimalist design' heuristic.

Conclusion

There are many positive things that can be said about ProfCast. It provides a quick and clean way to make a compact podcast of a PowerPoint or Keynote presentation. In this report, we focused on 29 violations of Jakob Nielsen's ten heuristics that we found while evaluating ProfCast. We feel that correcting these problems will make ProfCast an even stronger application.

References

Nielson, Jakob. *Heuristic Evaluation*. Retrieved March 7, 2007, from <http://www.useit.com/papers/heuristic/>