Human Computer Interaction 2011/2012
Coursework 2

Stephen McGruer (s0840449) and Wenqi Yao(s0838969)

November 18, 2011

1 Introduction

In this paper, we evaluate the application *Tag App* by comparing it against Nielsen's usability heuristics, as well as conducting cooperative and natural user testing. These results are combined with our previous evaluation of the MIT *LabelMe* interface to identify key strengths, weaknesses, and improvements for our own application *Image Labeller*.

2 Initial Evaluation

We first applied Nielsen's usability heuristics[1] to determine *Tag App*'s strong and weak areas, and to identify problems that users would likely come up against.

2.1 Key Strengths

One strength that we found was the visibility of changes in the system status - five out of the seven buttons gave clear visual feedback. The application also avoided using overly technical language, and supported recognition-based over recall-based interaction (avoiding hidden options or menus.) A help file was made available, with concrete steps for users to follow. Finally, it had a clean and minimalistic design with no unnecessary options, although the chosen colour scheme (orange and yellow) might not be appealing to all users.

2.2 Key Limitations

Placement of and feedback from buttons was not ideal. In particular, clicking the 'Save' button disabled it without confirmation of the system status (e.g. an 'image saved' pop-up). The 'Open' and 'Save' buttons, which affect the entire image, were mixed with buttons such as 'New', which only affects a single label. This is illustrated in **Figure 1**. Additionally, the 'Help' button, which plays a key role in user guidance, was not in an eye-catching position.

There was also a lack of consistency and accuracy in the use of language. Labels were referred to at different points as 'objects' and 'labels', while the act of adding a label was referred to as 'adding an object', 'creating a label', and 'tagging an object'. The 'Edit' button's actual function was to rename the selected label.

There was poor support for user control. Keyboard shortcuts or application preferences could not be set, even though letters in some names were underlined corresponding to the general standards for shortcuts (e.g. 'Open'). The short-

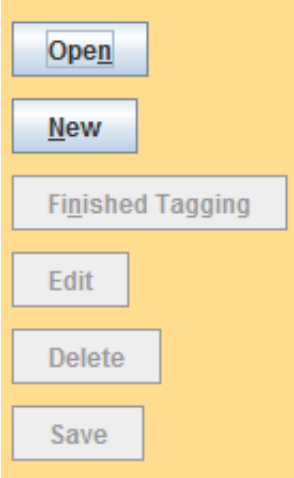

Figure 1: Button panel, showcasing inconsistent grouping. 'Help' is not on the button panel.

cut *Alt-N* opened the Help file, but this was never documented, suggesting that it was unintentional.

Finally, the application was inflexible towards many possible changes in users' decisions. Global undo/redo functionality was not available. Also, there was no option to cancel label creation after providing a name, so the only way to stop creating a label was to finish it. Right clicking while drawing a label would undo the last point; however, this functionality was not intuitive.

2.3 Bugs Found

- *Clicking the 'Delete' button while multiple labels were selected* would only delete one label. The 'Delete' and 'Edit' buttons would be left disabled, so other labels could not be deleted or renamed.

- *Clicking the 'Home' button in the 'Open Image' dialogue* would disable the 'up' arrow, leaving the dialogue stuck in the current folder.

3 User Evaluation

3.1 Methodology

We used two types of evaluation methods to test the software; Cooperative Evaluation and Natural Evaluation.

Selecting Subjects. Nielsen and Landauer suggest that using three to five subjects yields the maximum benefit to cost ratio[2]. However, Nielsen also argues that when novices evaluate an application, around 14 subjects are needed to identify 75% of problems [3]. As *Tag App* has limited functionality, we decided that a smaller number of subjects would not affect results greatly, and it was more important to maximise our cost-benefit. Hence, we used five subjects for cooperative evaluation, and four for natural evaluation. We selected a mix of male and female users from a variety of backgrounds to represent the intended user base. A short description of each user can be found in **Appendix A: User Descriptions**.

Procedure. Before testing the software, users were briefed on the aims of our project, as well as the rationale behind creating applications for labelling images. The 'Thinking-Aloud' technique was explained in order to facilitate dialogue between the user and experimenter, and to disambiguate users actions. Users were also told about the experiment procedure, and informed that they might withdraw their participation at any time. After the evaluation, users were asked for feedback on aspects of the program that confused them, and for suggestions on improving the interface.

Recording Results. Subjects' actions and comments were noted by hand. As stated in our previous report, differences in subjects' reading speed and memory of the tasks given would affect the time taken for them to accomplish each task. These timings were therefore not recorded. Quantitative data was instead taken from subjects' rating of the application, as well as the number of subjects who performed each action.

3.1.1 Cooperative Evaluation

Testing was carried out in small rooms in Appleton Tower on the University DICE machines. Users were given a list of six tasks to complete in order:

1. Labelling one object in a scene
2. Opening an image from a different folder
3. Labelling multiple objects in a scene
4. Deleting objects
5. Editing the area of a label
6. Closing the program

These six tasks covered all functionality in the program, except (1) undoing a point while creating a label, and (2) renaming a label. After each user completed Task 3, they were prompted by the experimenter to rename labels, through the use of simple scenarios (e.g. "What if a machine doesn't know that 'Hut 1' and 'Hut 2' are both the same kind of object?"). Not explicitly asking the users to carry out these tasks is in line with Molich's [4] testing guidelines, which states that users should be given goals, rather than steps or descriptions. The task sheet given to users can be found in **Appendix B: User Tasks**.

After completing the tasks, users were given a short survey about their experience completing each task. The questions from this survey can be found in **Appendix C: User Survey**.

3.1.2 Natural Evaluation

To make the evaluation setting resemble environments that a user would normally be in, the program was run on a Windows environment (rather than Linux) on laptops, in a range of different locations. Each user was provided with a sample image to label, but given free rein to import any images from other folders, as well as create any labels that they wanted to.

3.2 Results

3.2.1 Cooperative Evaluation

Subjects found it difficult to accomplish the given tasks, with one user describing the program as 'very demanding on users'. The subjects scored each task out of 5, with 1 being the lowest and 5 being the highest. They found it easiest to select (4.6pts) and delete (4.4pts) labels; while opening an image (2.6pts) was deemed the hardest task, followed by label editing (3.0pts) and then creating a label (3.2pts).

While all subjects were able to save and close the image, they ran into a multitude of other problems:

Unintuitive Interface. None of the subjects managed to create a label on their first try, or load an image without assistance. To create a label they attempted a variety of methods including clicking directly on the image (four subjects) and clicking the 'Open' button (two subjects). To load an image, every subject tried to click the location bar to change the directory, and three subjects tried to click the 'File Type'

field. Additionally, one subject tried to click the 'Home' button when opening a new image, locking the loading dialog in a state where buttons no longer worked. Finally, when creating more labels in Task 3, two subjects replicated mistakes that they had made earlier.

Poor Documentation. All subjects were prompted to check the help file when stuck, but this was not always useful. After reading the documentation, one subject tried to right-click to undo points, but was confused by the colours of the lines changing arbitrarily. Another subject did not understand the instruction 'add vertices to the image panel', as she did not know what the image panel was. Subjects also did not always use the same language as the help documentation - for example, one subject searched the section titled 'Delete Label' instead of looking for 'undo' as they wanted to 'delete the last line [they] drew'.

Bugs & Inflexibility. One subject made a spelling error in naming a label. Since they could not cancel or rename the label within the label creation state, they decided to 'make it and then delete it later'. Also, all of the subjects created multiple labels with the same name. Since all of these labels were highlighted when any were selected, two users accidentally deleted the wrong label.

3.2.2 Natural Evaluation

A 10 minute time frame was suggested for subjects to explore *Tag App*. On average, subjects spent 13 minutes and 20 seconds using the application, though this was skewed by an enthusiastic subject who used it for 20 minutes and 3 seconds.

The table below shows each feature that we identified in the program, and the number of subjects who successfully used it.

Feature	#
Opening a new image	3
Creating a new label	4
Undoing points during label creation	0
Renaming a label	2
Deleting a label	1
Saving an image	4
Using the help file	2

Table 1: # of subjects who used each feature.

Subjects also made several errors, mostly due to misconceptions about the program. These are detailed in the next table:

Error/Misconception	#
Changing folder to open image:	
Location/File Type bars clicked	4
'Home' icon clicked	2
Create new label -	
Start New: 'Open' clicked	2
Create Points: Cursor dragged instead of clicked	1
Edit label points: 'Edit' clicked	2
Accidents: Clicked 'Delete' for 'Edit'	1

Table 2: # of subjects who made each error.

Furthermore, subjects did not always understand the program output, as shown below.

Aspect causing confusion	#
Changing of label colours	2
Highlighting of selected labels	2
Vocabulary: 'vertices' for points	2
Multiple labels with the same name	2

Table 3: # of subjects confused by each aspect.

Gaps in Analysis. Even though we utilised

the 'Thinking-Aloud' method, it was not always possible to identify subjects intents. One subject commented 'I want to draw a box around the lamp', and promptly clicked the 'Open' button. It was unclear if two of the subjects meant to create a new label when they clicked 'New', as they were clicking the buttons systematically from top to bottom.

We were unable to test the usefulness of the help documentation. Two subjects did not click the 'Help' button, attributing this to 'never us[ing] the help file' when working with other programs. One subject only noticed it halfway during the evaluation, and the final subject did not find extra functionality from reading it.

3.2.3 Bugs

Two subjects found new bugs in the application.

- *Clicking the 'X' button on the 'Save Image?' dialog* acts as if you clicked 'No' rather than 'Cancel', so any subsequent dialogs (such as the Open Image dialog) appear anyway, without feedback about the state.
- *Opening a non-image file* crashes the application.

4 Discussion of Results

We found that our hypotheses (based on Nielsen's heuristics) were generally mirrored by the actions and thoughts of the subjects involved in our two evaluations. However, there were some cases where feedback did not match our predictions. In this section we discuss these discrepancies between theory and practice, and suggest corresponding improvements.

As expected, subjects seemed to seek out and respond to changes in the system state. They saw and interacted with pop-up dialogues displayed after they clicked a button, and were confused by the lack of feedback from 'Save' functionality, especially during the transition between the 'Do you want to save?' and 'Open Image' dialogues.

However, at least four subjects did not make the link between a label being highlighted and it being selected. This might be due to the change in various visual cues when a label is selected: the arbitrary changes in the colours of non-selected labels could detract from the thick, bright red line of a selected label. This problem could be reduced by tying each label to a particular colour, or colouring each label by its current state (e.g. Selected).

Lack of consistency did cause confusion around the act of 'labelling/tagging' and the button functionality. However, we failed to recognise several words that were used that might not be suitable for users. For example, a user was confused by a reference to the 'Image Panel' in the help documentation, as the meaning of this phrase was never defined. Ways to mitigate this include renaming the buttons (e.g. changing 'New' to 'New Label', 'Finished Tagging' to 'Finished Labelling') as well as including a help graphic which points out the names and locations of different panels.

Subjects had trouble repeating previous tasks that they had performed, showing that they had to recall action sequences rather than immediately recognise what to do next. One noted that this behaviour was due to the ambiguously named buttons. The renaming of buttons as suggested above would help to alleviate this problem.

Contrary to our expectations, subjects struggled to use the help documentation. Problems included finding the section in the help file that

was appropriate to their problem, and the general wording of the documentation. While three subjects used the help documentation to learn about the undo functionality, two others mentioned wanting to undo points, but did not find the corresponding section in the documentation. The documentation might benefit from an FAQ list, as there were many common questions such as 'How do I make a line?'

Subjects replicated many of the bugs that we found. One wanted to cancel the label that he had just started creating, but was unable to do so. This problem would be easily eliminated with the addition of a 'Cancel' button. Four subjects locked the 'Open Image' dialogue in a state where only the current folder could be accessed, since they could not understand many buttons and fields in the dialogue. No subjects triggered the multiple-label-delete error, possibly due to the limited time spent and labels made.

4.1 User Suggestions

Subjects in both evaluations made several suggestions, all of which could be a valuable addition to the application. However, due to development costs and characteristics of the intended user base, we found some improvements to be more critical than others.

4.1.1 Functional Aspects

One user thought that having the cursor hover over a label on the image should cause the name of the label to appear. This would increase the users mental link between a label and its name, making it easier to select a label from the list. Two users thought it would be useful to draw curves - we previously thought of this while developing our own application, but found it too difficult to implement. Two subjects opened images that were not ideally sized, prompting them to suggest zoom or resizing functionality. This would be a good addition to the program if unusually sized images were expected, although the high cost of implementing a zoom functionality means that we would suggest resizing images instead.

4.1.2 Linguistic aspects

One user mentioned that the button names were ambiguous, as 'New' could mean either 'New Label' or 'New Image', and suggested that we make them more specific. Additionally, two users were confused by the use of both 'labels' and 'tagging' in the application. As we have mentioned above, we think reducing the ambiguity in *Tag App* is very important, and as such would consider this a necessary change for the application. Another user was confused by the use of the word 'vertices' in the context of what she considered 'putting dots' (before she created enough to see a line). As only one user had problems with this use of language, we consider that 'vertices' is not too technical a word, and we would continue to use it.

5 Comparative Evaluation

To identify key improvements for our application *Image Labeller*, we compared it against both our new *Tag App* discoveries and a previous analytical evaluation of MITs *LabelMe*[1].

Image Labeller was inferior to other the applications in three main areas:

[1]See our previous report

Recognition-based Interfaces. In *Tag App* and *LabelMe*, all available actions are contained on the screen (with the exception of *Tag App*s 'Undo'). While the main actions within *Image Labeller* are always visible, others are hidden in the menu-bar and the user must remember which menu they are in.

Visual Links and Indicators. When a label is selected, *Tag App* highlights it with a thick red line and *LabelMe* shades the area contained by its edges. While *Image Labeller* does change the colour of the selected label, this is not as clear as the other two applications. When creating a label, a useful function in *LabelMe* is the large marker indicating the first point added. Neither *Tag App* nor *Image Labeller* offer such a feature.

Use of Technical Language. Most instructions in *Tag App* and *LabelMe* use simple, non-technical language. However, the concept of 'Collections' in *Image Labeller* may be considered too abstract for the novice user.

Some problems in *Tag App* and *LabelMe* revealed similar issues in *Image Labeller*:

Poor Help Documentation. Both *Tag App* and *Image Labeller* have help functions that are often overlooked. *LabelMe* uses its help documentation as a landing page, at the expense of making the actual application hard to find.

Ambiguous Buttons. All three applications suffer from some ambiguity in their button names; *Tag App* for almost all buttons, *LabelMe* in the 'Erase' button, and *Image Labeller* in its icon-based buttons.

Other problems in *Tag App* and *LabelMe* highlighted positive aspects of *Image Labeller*:

Display of Available Actions. In *Tag App*, one can click the 'Delete' button when more than one label is selected, even though only one label will be deleted. The labels that remain cannot be selected for deletion. *LabelMe* allows the user to click the 'Erase' button even when nothing can be erased. On the other hand, actions that might lead to the user being trapped in a particular state have been disabled in *Image Labeller*.

Based on our comparison, we came up with five changes we would make to Image Labeller:

Implementation of an Icon-Bar. Icon-based shortcuts for actions such as 'Save' would promote recognition-based interaction.

Shading of Selected Label Area. Increased visibility of selected labels would strengthen users' mental link between label names and corresponding areas.

Enlarged First Points in Label Creation. The lack of this feature did not hinder subjects during iterative development tests. However, the increased clarity from a larger first point would be well worth the small cost of implementation.

Addition of a Tutorial. As suggested by one of the subjects from our natural evaluation, users could be familiarised with the application with a tutorial that appeared on their first visit.

Reducing Button Ambiguity. Studies have shown that *user-preferred* names overlap by only 15-35% [5]. However, to maximise the clarity of our button names and icons, we would like to run a study to find the best *user-understood* names and pictures, even if these would not be the first-choice names for most users.

6 Conclusion

Nielsen's usability heuristics was effective in helping us identify many of *Tag App*'s limitations, as shown by the strong correlation between

aspects we highlighted and issues faced by real users during testing. However, results from user evaluation were crucial in aiding our understanding of the extent of each problem. With this deeper knowledge of the impact of strengths and limitations in *Tag App* on real users, it was possible to identify a set of improvements in *Image Labeller* that would optimise user benefit while minimising the extra cost of development.

References

[1] Nielsen, J., *Enhancing the explanatory power of usability heuristics.* Proceedings of the ACM CHI'94 Conference, 1994

[2] Nielsen, J., and Landauer, T.K., *A mathematical model of the finding of usability problems.* Proceedings of ACM INTERCHI'93 Conference, pp. 206-213, 1993

[3] Nielsen, J., *Usability Engineering.* Academic Press, pp. 156, 1993

[4] Molich, R., *230 Tips and Tricks for a Better Usability Test.* Nielsen Norman Group, 2001

[5] Furnas, G. W., Landauer, T. K., Gomez, L. M., Dumais, S. T., *The Vocabulary Problem in Human-System Communication.* Communications of the ACM, 30 (11), 964-971., 1987

Appendices

A User Descriptions

A.1 Cooperative Evaluation

- User 1 is a 22 year old female and a 4th year Biology graduate.
- User 2 is a 20 year old female and a 2nd year undergraduate Law student.
- User 3 is an 18 year old female and a 1st year undergraduate Politics with Geography student.
- User 4 is a 22 year old male and a postgraduate Linguistics student.
- User 5 is a 22 year old female working in biomedical research.

A.2 Natural Evaluation

- User A is a 22 year old male and a 3rd year undergraduate Pharmacology student. Testing was carried out at a desk in a bedroom.
- User B is a 21 year old female and a 4th year undergraduate History student. Testing was carried at a sofa in an open area of Appleton Tower.
- User C is a 21 year old female and a 4th year undergraduate Spanish and English Literature student. Testing was carried out at a desk in room 5.07 of Appleton Tower.
- User D is a 22 year old male and a Biology graduate working in a watch store. Testing was carried out at a desk in his home.

B User Tasks

User Evaluation

In this evaluation, you will be testing some software for labelling images (i.e. for indicating what region of an image contains what objects). You will be asked to label a few images, and afterwards will be asked a few general questions about your opinion on the software.

Task #1
You should see a picture of a Hut.

- Please label the hut "Hut".
- Save the image.

Task #2 *Im a nut, thats not a real hut!*
You are currently in the folder **/HCI/Eval/images**.

- Please open the image "huts.jpg" from the folder **/HCI/Images**.

Task #3 *A machine will need some luck to find these hut[s]...*

- Please label every hut in the image you have just loaded.

Task #4 *One hut (or more) wont make your cut.*

- Please delete the 2 huts you dislike the most.

Task #5 *Did we forget a little jut?*

- Please reload the first image, "hut.jpg" from the folder **/HCI/Eval/images**.
- Ensure that your label covers the signboard outside the store.

Task #6

- Please close the program.

C User Survey

This survey was given as an online form. Users were asked to rate each aspect of the program on a scale of 1 to 5, where 5 was the best score (e.g. task was easiest to accomplish).

C.1 Survey Questions

1. How easy was it to OPEN an image?
2. How easy was it to CREATE a label?
3. How easy was it to SELECT a label?
4. How easy was it to EDIT a label?
5. How easy was it to DELETE a label?
6. How AESTHETICALLY PLEASING was the software?

C.2 Online Survey

The online survey can be found on:
`http://docs.google.com/spreadsheet/viewform?hl=en_US&formkey=dG1PSWZQX1M0ZkFrNVJ4WXhHcm5CTEE6MQ#gid=0`