# An Empirical Evaluation of Collapsible Panel Interfaces

Joshua Leung
*Department of Computer Science and Software Engineering*
*University of Canterbury*
Christchurch, New Zealand
joshua.leung@canterbury.ac.nz

Andy Cockburn
*Department of Computer Science and Software Engineering*
*University of Canterbury*
Christchurch, New Zealand
andy@cosc.canterbury.ac.nz

***Abstract*—Although collapsible panel widgets have been widely used in industry for many years in various contexts, there have been no formal evaluations conducted to compare user performance and satisfaction with these interfaces. There is also a lack of formal guidance for user interface designers about how to choose among the many variations of collapsible panel schemes in existence. In this paper, we develop a classification framework and propose an empirical method for evaluating user performance with these interfaces. An experiment was conducted comparing accordions, collapsible panels, and flat panel layouts. This found a clear preference for accordions, with the need to scroll to find targets in the other interfaces cited as a key disadvantage of those approaches. Implications of these findings are discussed.**

***Index Terms*—accordion, collapsible panels, empirical evaluation, hierarchical list, graphical user interface, information architecture, user interface design patterns**

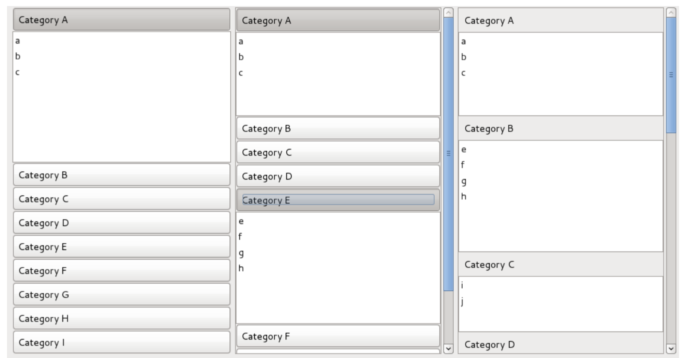


Fig. 1. Examples of collapsible panel schemes – a) Accordion, b) Collapsible Panels, c) Flat Panels (non-collapsible)

## I. Introduction

Modern graphical user interfaces are often large and complicated systems. They contain hundreds of tools and options that users need and expect to be able to easily access as part of their daily workflows. Hierarchical information architectures are used to help users deal with this complexity, by grouping the items in a structured manner. Examples of widgets used for this purpose include menus, tabs, the Microsoft "Ribbon", task panes, and collapsible panel schemes.

Figure 1 shows several examples of "collapsible panel" widgets, but also a non-collapsible panels scheme. In each case, there is a vertical (or horizontal) stack of panels, where each panel consists of a labelled header and an adjoining region where the category's items are displayed. The main difference between flat (or static) panel layouts and collapsible panels is the ability to use the category labels of collapsible panel interfaces to elide the associated contents as appropriate.

A common use case for collapsible panel widgets is for "Tool Palette" or "Gallery" panes for content creation tasks (such as when drawing diagrams [12], typesetting music [10], or laying out widgets on a form [8]). In these contexts, the panels on the side of the workspace to provide quick and easy access to a large number of shapes or effects. They have also been used as navigational schemes for web sites and for mobile devices with limited screen sizes [1].

Some of these latter uses can be attributed to the "Accordion" User Interface Design Pattern (UIDP) [11]. Inspired by the "Design Patterns" movement in Software Engineering [5], UIDP's aim to document the context, rationale, and solution to a design problem that is commonly encountered [6]. It is important to note that although design patterns describe a solution to a common problem, they do not prescribe specific concrete instantiations of the proposed solution, but rather a class of approaches to be considered.

Although there have been some prior studies of interfaces with hierarchical structures, there is a lack of any research or publicly available data comparing different collapsible panel interfaces. Hence, there is a lack of any guiding principles or empirical data illustrating important considerations for designers when choosing appropriate widgets.

To address these concerns, we develop a classification scheme for collapsible panel schemes, and describe an experimental method for characterising user performance and preferences for representative interfaces from each category. Results from our initial results are presented and used to provide qualified guidance to user interface designers on the relative merits of various designs.

## II. Related Work

To the best of our knowledge, there has not been any published research directly comparing user performance with different collapsible panel schemes resembling the interfaces described here. Prior studies have however been performed for other hierarchical structures such as menus or tree-based "table of contents" displays.

### A. *Performance of Menus*

User performance with menu structures has been well studied. [4] developed a performance model for menu use, which combines terms modelling visual search time (Fitts' Law) and 'optimally prepared' choice reaction time (Hick-Hyman's Law). This model is based on the idea that while novice users initially have to spend time visually searching for items, more experienced users learn the locations of targets.

[9] evaluated a menu design where all menus were shown in parallel at the same time, thus flattening the hierarchy. Their results showed that task performance time was reduced using this interface, suggesting that it may be beneficial to not require categories to be expanded before grouped items can be located.

### B. *Hierarchical Content - Trees and Lists*

[2] performed a study evaluating three different interfaces for navigating large Table of Contents trees which contained 1296 items nested three levels deep.

The interfaces studied were a fully expanded tree (equivalent to the 'flat' condition in Figure 1c), a tree where individual levels could be expanded/collapsed independently (equivalent to the 'collapsible panels' condition in Figure 1b), and an interface where each depth level was represented as a separate panel resulting in a split-levels approach which the authors referred to as the "multi-pane" approach. It was found that collapsible levels and multi-pane approach outperformed the fully expanded tree, with the multi-pane approach being the most efficient overall.

[3] developed a performance model for navigation of hierarchical lists by scrolling. This study proposes that user performance with hierarchical list structures depends on their ability to predict the location of items and their familiarity with the classification and structure of the information presented.

## III. Industrial Usage of Collapsible Panels

We analysed the usage of collapsible panel "tool palette" components in the following commonly-used applications: yEd [12], LucidCharts [7], MuseScore 1.2 [10], and Qt Creator [8].

Each set of collapsible panels in the applications listed were counted. Where applications had panels containing panels which could be dynamically populated with user content, only the system-defined items were included.

### A. *Palette Survey Results*

The number of panels (Table I) appears to follow a normal distribution with centered around 11-12 items. Overall, panel sizes (number of items in each panel) appear to follow an exponential distribution, although there seem to be two distinct 'peaks'. Most panels are small, with an average of 10 items per panel. However, there are a very small (1-2) number of large panels which have greater than 30 items each.

## IV. Types of Collapsible Panels

From our analysis of existing industrial usage of collapsible panel schemes, we identified three design parameters which define the behaviour of most collapsible panel interfaces:

$$\begin{aligned}\text{Collapsibility of Panels} &\in \{\text{Flat, Collapsible Panels}\}\\ \text{Panels Expandable} &\in \{\text{Single, Multiple}\}\\ \text{Category Placement} &\in \{\text{Fixed, Non-Scrolling, Scrolling}\}\end{aligned}$$

Of these factors, the *Category Placement* factor is perhaps the least intuitive. This factor concerns the position and size of panels relative to the bounds of the host container. The *Fixed* and *Non-Scrolling* types ensure that all category labels are always visible (i.e. without scrolling). With *Fixed* panels, there must always be at least one panel open, and the collapsed labels are always packed against the top and bottom edges of the host by stretching the open panel(s) as necessary.

Additional factors such as the presence and/or length of animated transitions between states are not included in this classification scheme, as they can be equally applied across all interfaces without affecting the fundamental nature of how the widget responds to input.

### A. *Example Classifications*

Using this classification, the basic "*Accordion*" (or "Outlook Bar") interface is a Single Fixed Collapsible scheme. This is because there must only be a single panel open at a time, and all category labels are packed so that they are always visible within the bounds of the host container. The selected panel expands to fill the space in the center of the container, with the other category labels stacked above and below this.

The "*Collapsible Panel*" scheme used in this paper is a Multiple Scrolling Collapsible scheme.

## V. Study Method

### A. *Goals*

1) Determine whether having panels which can be collapsed is more efficient than panels which are always open
2) Determine whether the restriction of only having a single panel open is detrimental to performance
3) Determine whether always keeping all category labels in view is beneficial
4) Characterise the scalability of these interfaces with regard to number of categories of items to display
5) Identify which of these interfaces is most preferred and/or most efficient to use

### B. *Experiment Design*

A $3 \times 3$, within-subjects design was used to evaluate the effectiveness of three representative interfaces at three different data (category) densities. For each condition investigated, participants were required to perform 6 repetitions.

The factors investigated were:

$$\begin{aligned}\text{Interface Type} &\in \{\text{Accordion, Collapsible Panels, Flat}\}\\ \text{Category Density} &\in \{\text{Low (5), Medium (11), High (20)}\}\end{aligned}$$

TABLE I
RESULTS OF SURVEY OF COLLAPSIBLE PANEL INTERFACES IN EXISTING SOFTWARE

| Application | Interface Type | Num Open | Num Panels | Mean Items | Stdev Items | Min | Max | Total Items |
|---|---|---|---|---|---|---|---|---|
| yEd | Multi Scroll | *1* | 12 | 15.63 | 10.94 | 2 | 36 | 172 |
| LucidCharts | Multi Scroll | All | 10 | 8.1 | 6.44 | 1 | 23 | 81 |
| MuseScore | Multi *Non-Scrolling* | 0 | 20 | 10.85 | 8.73 | 1 | 35 | 217 |
| QtCreator | Multi Scroll | All | 8 | 6.5 | 4.31 | 2 | 15 | 52 |

Ordering effects were controlled/counterbalanced by using permutations of the three interface types, so that participants were not all experiencing the same interfaces in the same order. However, the order for category densities were not formally counterbalanced due to study size limitations, but were instead randomly chosen at the start of each block. The targets for each task were determined randomly.

### C. Participants

Participants were postgraduate students enrolled in a computer science course at a local university, and were participating in experiments as part of their coursework. There were 9 participants in this experiment (8 male, 1 female), chosen on a semi-random basis (i.e. by availability during the experimentation period). All were frequent computer users.

### D. Apparatus

All experiments were conducted on a workstation running the Gnome window manager on Fedora 16, and displayed on a 22 inch monitor at a resolution of $1680 \times 1050$. A Microsoft three-button optical mouse was used for pointing.

The experimental interface (shown in Figure 2) was implemented using PyQt4. It consisted of a single window complete with menu and status bars, and displayed full-screen to minimise distraction and simulate the typical operating conditions of a standard content creation application. The collapsible panel interface for each condition was displayed in a 'Palette' side panel spanning the entire height of the window.

The task prompter was located in the main area (to the right of the 'Palette') where the primary content (e.g. canvas) of the application would usually appear. Participants were shown the *Category* and *Item Name* of the target item in large font.

The collapsible panel interfaces were hand-built and instrumented to log important events such as clicks and double-clicks down to millisecond accuracy, to be analysed after the experiment to obtain the task times and other measures. Correctly double-clicking on the targets would trigger log messages for ending the previous task's timing period but also starting the next time period. Task times were calculated from these start/stop events for tasks.

### E. Procedure

At the start of the experiment, the task was verbally explained to participants, with the relationship between the task prompting area and the experimental interfaces explicitly pointed out to avoid confusion. Participants were told to carry out the tasks as quickly as possible.

The experiment was conducted in three blocks, where each participant would complete all the tasks with one interface, and then given an opportunity to take a short break before the trials for the next interface were conducted. Participants were told that the interface behaviour had changed before the trials for the next block were performed.

For each trial within a block, participants were required to locate and double-click on the required targets. Between conditions (i.e. whenever the interface type or the number of categories changed), the collapsible panels interface was rebuilt and reset to its default configuration for the new condition size.

At the conclusion of the experiment, participants were verbally asked to rank the interfaces in order of preference (from best to worst). They were then asked to elaborate on the reasons for their choices, and to comment about any other issues they may have noticed or would like to mention about the interfaces they experienced.

## VI. RESULTS

### A. User Performance

Task times were calculated from the event logs by extracting the timestamps for pairs of task start/end events and calculating the time difference. To check for and eliminate bias from task acclimatisation effects at the start of each condition, the task times were evaluated using a two-factor ANOVA analysis twice – the first time with the full dataset (Figure 3), and the second time with the first trial of each condition removed (Figure 4).

Overall, there was a reduction in mean task times when these initial trials were removed, suggesting there was an acclimatisation effect. This could have been due to participants needing to move the cursor from the "Start Timing" button in the task prompter over to the "Palette" area at the start of each block. The rest of the results reported here therefore focus on the results with these initial trials removed (Figure 4).

As the graphs show, significant effects were detected for both *Interface Type* ($F_{2,16} = 8.346, p < 0.01$) and *Number of Categories* ($F_{2,16} = 7.852, p < 0.01$) factors studied. *Accordions* consistently outperformed *Collapsible Panels*, with mean times per selection of 3.262 sec ($SE = 0.1285$) and 3.778 sec ($SE = 0.1523$) respectively. *Flat Panels* were the least scalable approach, with a mean selection time of 4.319 sec ($SE = 0.2063$).

Further (visual) analysis of trends in the data reveals that *Accordions* and *Collapsible Panels* respond to category size in a similar way, with similarly-shaped response curves with increasing category density. The main difference between

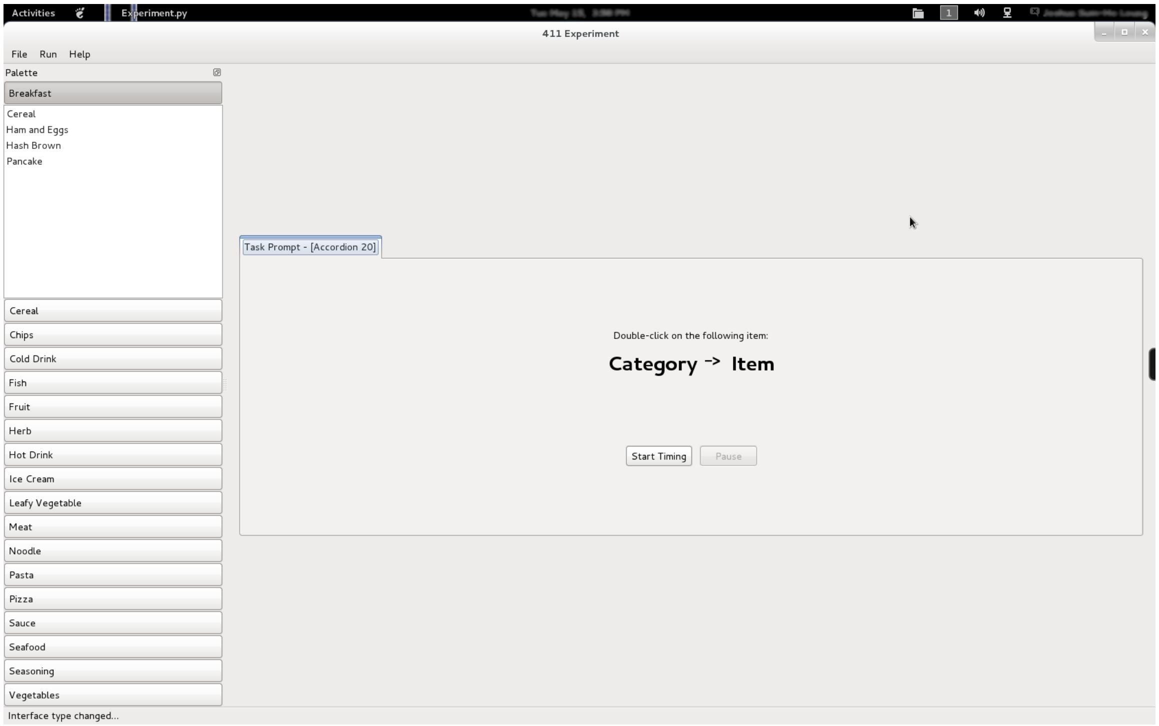


Fig. 2. Screenshot of experiment setup. The collapsible panel interfaces are displayed in the 'Palette' on the left. Tasks are displayed in the main area to the right of this.

these is that *Collapsible Panels* are less efficient overall, which could be attributed to the increased scrolling that participants performed to navigate between open panels. It was also observed that panel management behaviours such as closing panels after use, or repeatedly toggling a panel in confusion, could also have accounted for some performance degradation with Collapsible Panels. However, the extent of such behaviours appeared to vary between participants, which may have contributed to the larger amount of variation in the data for Collapsible Panels.

Compared to the other two interfaces, *Flat Panels* did not appear to be scalable with larger category sizes, with significantly higher task times recorded (4.731 and 4.809 seconds compared with less than 4 seconds for the other interfaces). This difference appears to be caused by the need to scroll long distances to jump between targets, compared to simply moving the mouse a fixed distance and clicking.

With the lowest category density studied, *Flat Panels* appeared to be slightly more efficient than *Collapsible Panels*, as participants did not have to first expand the categories to select the items. The graph shows that a crossover occurs here, which means that there is a significant interaction between factors here.

Overall, the response curve of the *Flat Panels* interface appears to follow a logarithmic pattern, although additional datapoints are needed to conclusively prove this. According to [3], this may indicate that participants were likely relying on some domain and/or spatial knowledge of the layout (possibly the alphabetical ordering) to locate items instead of performing a blind visual search.

Curiously, user performance with all interfaces appeared to improve with the highest category density. This was most pronounced for *Accordions* and *Collapsible Panels*, where the high density mean times are lower than those for medium density, but still higher than for the lowest density. The cause of this effect is not clear, but may have resulted from structural properties of the dataset used that this experiment did not evaluate.

### B. *Preferences and Subjective Evaluations*

When asked to rank the interfaces in order of preference, participants' responses fell into two groups:

1) *Group 1* – 7/9 of the participants responded that *Accordions* was their most favoured interface, followed by *Collapsible Panels*, and *Flat Panels* as the least preferred interface.
2) *Group 2* – The other (2/9) participants ranked the interfaces in the reverse order – *Flat Panels* as most preferred, then *Collapsible Panels*, and *Accordions* as least preferred

All participants in *Group 1* cited the need for "lots of scrolling" with the *Flat Panels* interface as the reason why it was the least preferred. One participant was heard exclaiming, "oh god!" upon encountering the *Flat Panels* high density condition.

*Collapsible Panels* were ranked as *Group 1*'s second most preferred option as, despite the need for scrolling, several (3/7) participants mentioned that the ability to choose when to

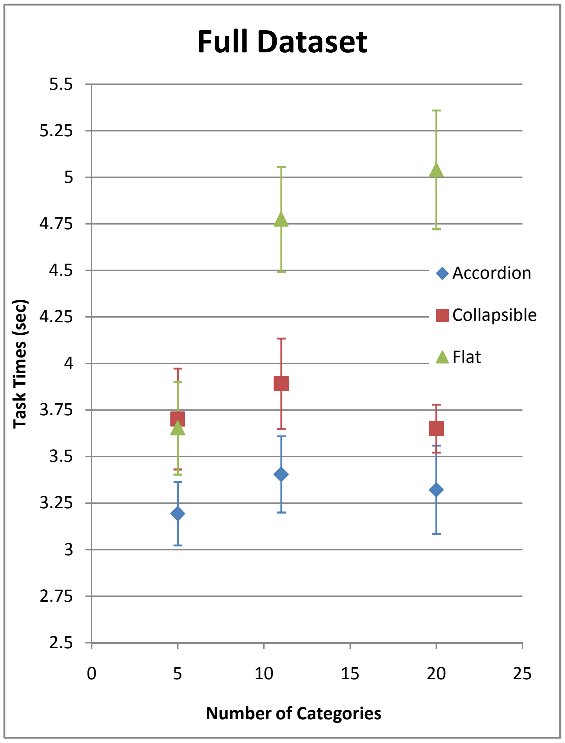


Fig. 3. Mean task times for Accordions, Collapsible Panels, and Flat Panels across a range of category densities (lower is better), with all data points included. All times in seconds.

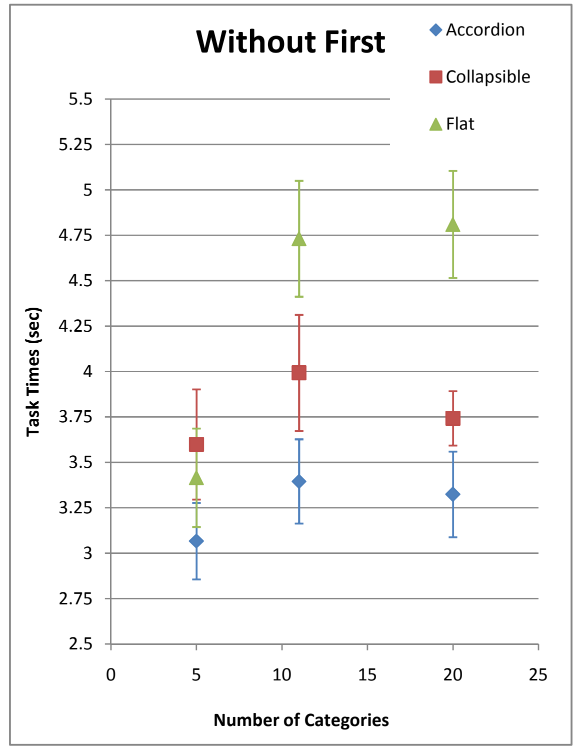


Fig. 4. Mean task times for Accordions, Collapsible Panels, and Flat Panels across a range of category densities (lower is better), with initial data points removed. All times in seconds.

open/close panels could be useful when frequently revisiting certain panels as this provided a form of organisation or control over the interface as needed for the current task. However, the lack of automatic collapsing was also cited by a few (2/7) participants as being a disadvantage of this approach.

*Accordions* were ranked as *Group 1*'s most preferred option. Common justifications included praise for the auto-collapsing behaviour reducing the need for manual management of panels, and not needing to scroll to find categories. Another interesting response was that items were "right there when you clicked on a category without having to hunt for the list". This suggests that when users have a specific target, they are more inclined to prefer a single-task-focus interface similar to traditional menus.

In contrast, participants in *Group 2* rated *Accordions* poorly due to the relative unpredictability of category label positions and the large amount of vertical mouse movement needed to toggle between panels. *Group 2* participants automatically reached for the scrollbar in the *Flat Panels* scenarios, whereas *Group 1* participants defaulted to clawing at the scroll wheel. This may explain one participant's comment that the *Flat Panels* interface was 'easy to use' because "you could just scroll and click".

## VII. Discussion

### A. Datasets and Content-Driven Interfaces

As highlighted by [3], user familiarity with the dataset and classification system used has a strong influence on task times with hierarchical structures.

The *Low* and *Medium* density datasets used here contained subsets of the full (*High*) dataset's panels. This improved internal validity as the different conditions were compared using relatively similar datasets, at the expense of being able to directly transfer findings to other datasets.

However, it is possible that structural characteristics (e.g. distribution and grouping of items) may have biased the results. For instance, one participant commented on the classifications for several items, while several others sometimes tried to find items in the wrong categories despite having been told where to look (via the prompt).

### B. Comparisons to Related Studies

Many similarities can be drawn between our study and [2], despite the differences in interfaces and contexts studied. Whereas the *Table of Contents* study investigated user performance with large three-level deep nested datasets, our study investigated user performance with shallower category $\rightarrow$ grouped-item interfaces.

In both studies, 'flat' or 'stable hierarchies' were found to be the least effective interfaces due to the need for users to scroll large distances to find targets.

Similarly, the collapsible panel interfaces, where individual categories can be expanded/collapsed manually by the user, were found to be significantly better in many cases, although the need for "housekeeping" (i.e. Section 5.1 "Practitioners Summary" from [2]) was cited as the primary disadvantage of this approach.

Thus, we conclude that collapsible panel interfaces can be considered a specialised sub-case of the more general Table of Contents hierarchical browsing problem.

### C. Validity of Tasks Performed

During the experiment, participants were required to find randomly selected targets. This was done as a crude simulation of general task performance with these interfaces, where items from different parts (some from the top, some from the bottom, some from categories in-between) of the interface may potentially need to be accessed in a similar fashion.

However, in practice, access patterns are highly dependent on both the user's tasks and the organisation of the panels (e.g. users may only repeatedly access a small subset of the entries in a typical palette). The method used here is a compromise between trying to model these behaviours and potentially over-constrain the interaction patterns to a particular optimal/worse case, or under-constrain the experiment compromising repeatability. It was also a compromise to ensure that the experiment length stayed within a reasonable timeframe for participation.

### D. Implications for Designers

From our results, *Accordion* interfaces appear to be the most scalable, efficient, and preferred interface among the interfaces evaluated, particularly for one-off or random-access patterns. If choosing between using *Accordions* and *Collapsible Panels*, *Accordions* appear to be better suited when users are aiming for a specific target within a category without needing to refer to another category, although some users may not like the additional mouse movement needed or the movement of category labels around the screen.

When a small number of categories is to be used, *Flat Panels* should be used in preference of *Collapsible Panels* due to the overheads associated with firstly opening the panels to find the relevant content. It appears that the optimal threshold for changing from *Flat Panels* to *Collapsible Panels* is around 6-7 panels. However, the sizes of individual panels may affect this figure, and needs to be investigated in further experiments.

### E. Future Work

Due to time constraints, we were only able to evaluate three different interfaces in this experiment. It would be interesting to compare other designs from the classification, such as Multiple Non-Scrolling Collapsible (e.g. "Multiple Accordions" [10]), Single Scrolling Collapsible, and split-level schemes (similar to the multi-pane approach in [2]).

The effects of category size and structure also need to be examined and characterised, along with the role of revisitation.

## VIII. Conclusion

In this paper, we have proposed a classification scheme for collapsible panels based on behavioural factors such as number of panels open and the visibility of category labels. An empirical method for evaluating and comparing the effectiveness (lowest task completion times, and subjective satisfaction) of such interfaces was described, and the results from an experiment comparing accordions, collapsible panels, and flat panel interfaces with 9 participants were presented and discussed. Accordion interfaces were found to be the most effective with average task completion times of 3.262 seconds and 78% of participants ranking this as their most preferred interface. The need to scroll long distances was cited as the primary annoyance with the other interfaces.

## References


[1] O. Buyukkokten, H. Garcia-molina, and A. Paepcke, "Accordion summarization for end-game browsing on PDAs and cellular phones," 2001, pp. 213–220.

[2] R. Chimera and B. Shneiderman, "An exploratory evaluation of three interfaces for browsing large hierarchical tables of contents," *ACM Transactions on Information Systems*, vol. 12, pp. 383–406, 1994.

[3] A. Cockburn and C. Gutwin, "A predictive model of human performance with scrolling and hierarchical lists," *Human—Computer Interaction*, vol. 24, no. 3, pp. 273–314, 2009.

[4] A. Cockburn, C. Gutwin, and S. Greenberg, "A predictive model of menu performance," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '07, 2007, pp. 627–636.

[5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. Addison-Wesley Professional, Nov 1994.

[6] J. Janeiro, S. D. J. Barbosa, T. Springer, and A. Schill, "A flexible model for improving the reuse of user interface design patterns," in *SIGDOC*, 2010, pp. 215–221.

[7] Lucid Software, "Lucidcharts," May 2012. [Online]. Available: http://www.lucidcharts.org

[8] Qt Company, "QtCreator – 1.3.1," February 2010. [Online]. Available: https://www.qt.io/product/development-tools

[9] P. Quinn and A. Cockburn, "The effects of menu parallelism on visual search and selection," in *Proceedings of the ninth conference on Australasian user interface - Volume 76*, ser. AUIC '08, 2008, pp. 79–84.

[10] W. Schweer, "MuseScore 1.2," 2012. [Online]. Available: http://www.musescore.org

[11] M. van Weile, "Welie.com - Patterns in interaction design," July 2009. [Online]. Available: http://www.welie.com/patterns/showPattern.php?patternID=accordion

[12] yWorks, "yEd Graph Editor," 2012. [Online]. Available: https://www.yworks.com/products/yed