Specifying Preferences Based On User History

Loren Terveen, Jessica McMackin, Brian Amento, Will Hill

AT&T Labs - Research
180 Park Avenue, P.O. Box 971
Florham Park, NJ 07932 USA
{terveen, brian, mcmackin, willhill}@research.att.com

Abstract

Many applications require users to specify preferences. We support users in this task by letting them define preferences relative to their personal history or that of other users. We implement this idea using a graphical technique called *control shadows*, which we have implemented on both a desktop computer and on a cell phone with a small, grayscale display. An empirical study compared user performance on the graphical interface and a text table interface with identical functionality. On the desktop, users completed their tasks more quickly and effectively and strongly preferred the graphical interface. On the cell phone, there was no significant difference between the graphical and table interfaces. Finally, personal history proved useful in specifying preferences, but history of other users was not helpful.

Keywords

history, reuse, visualization, collaborative filtering, mobile devices.

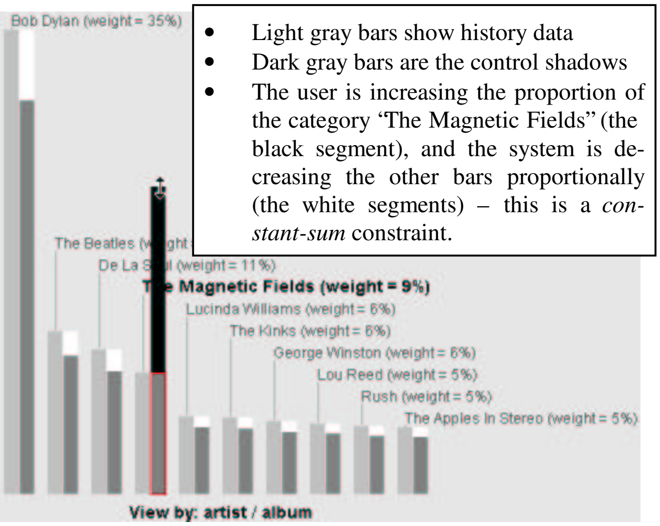
INTRODUCTION

In many interactive systems, users must specify a course of action to be implemented. A simple example is specifying types of news stories for a personal news service to deliver to you. Another example is specifying a playlist – a list of songs for a media player to play. A third example is defining a budget – allocating sums of money across a set of expense categories.

Interaction history is a prominent topic in HCI. The motivation is that behavior often remains quite consistent over time. If I listened to Bob Dylan, The Ramones, and The Mekons last week, I'm likely to want to listen to them this week, too. History data also may be shared: I might know several of my friends have good taste in music, so I ask them what they've been listening to lately. Such sharing of preferences is the foundation of collaborative filtering [13].

Our approach is to simplify the expression of complex specifications by letting users specify their intent relative to their personal history or the histories of other users. The rationale is that it is easier to modify an existing specification than to create a new one from scratch.

Control shadows are a graphical technique enabling history-based specification of intent. Display objects, such as the bars in a histogram, are given visual "shadows", e.g., a second set of bars. The shadows are active controls that enable users to express their intent in terms of the history data, stretching a bar to indicate "more of this" or shrinking a bar to indicate "less of that".


Bob Dylan (weight = 35%)
The Beatles (weight
De La Soul (weight = 11%)
The Magnetic Fields (weight = 9%)
Lucinda Williams (weight = 6%)
The Kinks (weight = 6%)
George Winston (weight = 6%)
Lou Reed (weight = 5%)
Rush (weight = 5%)
The Apples In Stereo (weight = 5%)
View by: artist / album

- Light gray bars show history data
- Dark gray bars are the control shadows
- The user is increasing the proportion of the category 'The Magnetic Fields" (the black segment), and the system is decreasing the other bars proportionally (the white segments) – this is a *constant-sum* constraint.

Figure 1: The control shadows graphical technique

We have implemented the control shadows technique in the *HistView* system. HistView also embodies a *constant-sum* constraint: as a user increases the importance of one category in the specification (for example, asks for more music by a particular artist), the system proportionally decreases the other categories (see Figure 1).

The control shadows technique is both simple and general. It is simple enough to be implemented on a cell phone with limited display and input capabilities. It is general enough to apply to many resource allocation problems, where a fixed resource (like disk space) must be allocated among a set of items (such as a set of TV programs to be recorded).

The rest of this paper is organized as follows. We first discuss related work. We then describe and illustrate the designs of both the desktop computer and cell phone systems. We next describe our evaluation experiment, present the results, and discuss design implications. We close with a brief summary of contributions.

CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA.

RELATED WORK

We compare our approach to related work in history reuse, collaborative filtering, and visualization/visual querying.

Early HCI research [6] showed that users of command line interfaces tended to use a small number of commands over and over. Building on this finding, systems began to record command histories and provide interfaces for viewing histories and selecting commands. Later Lee [10] and Greenberg [5] presented analyses of different ways in which interaction history could be used. Our approach is most closely related to Lee's *history for reuse*. However, rather than focus on reusing single history items, we let users identify patterns in history data and express new specifications based on these patterns.

The "history-enriched digital objects" line of work [7], [9] was based on the observation that real world objects accumulate *wear* through use, and that this wear — such as smudges on certain recipes in a cookbook — informs future usage. Systems were built that captured interaction history data such as reading time and used the data to modify the appearance of interface objects to reflect their use. For example, scrollbars were annotated with horizontal lines of differing length and color to represent amount of reading or editing by various users.

S*ocial navigation* research [12] builds on this foundation, using history to aid *navigation*, i.e., to help users decide where to go next in a virtual space. Footprints [16] records and analyzes user web browsing history, and constructs several visualizations to aid user navigation through a web site. Other collaborative filtering research aims to support people in sharing preferences. Maltz and Ehrlich [11] designed a system that lets users create digests of pointers to documents for sharing with others. However, such systems don't enable users to consult their history as part of sharing preferences. We believe that using history for this purpose is natural and simplifies the task. Our techniques let users examine their browsing or listening history to identify documents or songs they want to share.

Information visualization systems allow users to view and interact with large datasets. For example, SeeSoft [1] lets users explore large datasets such as code bases. Each file in the code base is represented by a vertical bar whose height is proportional to the file size. Bars are visually divided, with thin horizontal lines representing lines of source code in the files. The color of a line represents a property of the line in the source code, e.g., its age or the programmer who wrote it. SeeSoft helps users identify patterns such as which modules in a system have been modified a lot recently or which programmer is best to consult about a particular file. HistView also visualizes data as a series of bars, which are further subdivided. The visualizations differ in many details; however, more important, SeeSoft only visualizes data, while HistView lets users express new specifications based on history data.

Herndon et al [7] described what they called *interactive shadows* for computer graphics. Objects in a 3D environment were given literal shadows, projected onto a 2D plane. These shadows could be used to control the objects, e.g., to translate, rotate, and scale them.

Dynamic querying [14] turns data querying into a direct manipulation task. Queries are defined and modified by manipulating widgets such as sliders, there is instant visual feedback, and query manipulation is closely coupled with visualization of results. The Alphaslider [1] is a compact and efficient means of selecting items from a list. Eick [4] and Tweedie et al [15] explored ways of visualizing rich amounts of data directly on control widgets: Eick used a slider, and Tweedie et al used a type of interactive histogram. More recently, Wittenburg, Lanning, and colleagues [17] have experimented with interactive "bargrams" (derived from a histogram by "tipping" the bars horizontally and laying them end-to-end).

In our system, part of what users must do is select history data to base their specification on; for this part of the task, our system takes a dynamic querying approach. There are a number of differences, though. First, our system supports not just visualizing existing data, but also defining new specifications based on that data. Second, while the control shadows technique also tightly integrates visualization and control, it is less powerful, but more direct than the methods of Eick and Tweedie: bars represent quantities, and the quantities are changed by stretching the bars. Also, the object being controlled is different: category weights rather than query parameters. Further, techniques like the Alphaslider support selection of items from a list, while we emphasize setting parameters (category weights). And finding individual items is less important in our case; we are more concerned with letting users find and reuse patterns.

EXAMPLE DATASET

HistView processes history data represented as flat tables. In this paper, we use the example of history data from a shared music listening environment in use at AT&T. (HistView also has been applied to budgeting and news headline data.) Users submit tracks from a shared repository or their personal MP3 collections to the environment. The data from this application has the following attributes:

time	*// the date and time a track was played*
dj	*// the user who played this track*
track	
album	
artist	

USER TASKS

In our history-based specification approach, there are three main user tasks that any software must support.

1. *Forming categories* to organize and view the data. Categories are defined by selecting an attribute: for example, selecting the *dj* attribute means that there will be

one category for each person who has played a track in the listening history dataset. This gives a flavor of collaborative filtering: I may create a specification based on the listening histories of several of my friends whose musical taste I admire.

2. *Selecting categories* to include in or exclude from one's specification. In most cases, users want to base their specifications on only part of the history data. For example, if I'm a hip-hop fan, but I'm starting from a listening history with lots of classic rock in it, I'll quickly exclude artists like The Beatles, The Rolling Stones, Led Zeppelin, etc.
3. *Specifying the relative importance* of categories in the specification. As I add categories to my specification, I need to set their weight: it's as if I'm defining my own personal radio station that plays only a certain set of artists, and I need to specify what the mix is. For example, I might want to listen to Moby about 15% of the time, Madonna 10%, etc.

Note that a specification can be recursive – after selecting a set of artists to listen to, one may select particular albums by those artists, and even the tracks on these albums.

SYSTEM DESCRIPTION

The initial system implementation of HistView was in Java/Swing, intended to run on a standard desktop PC with a windowing operating system. Subsequently, we implemented the same functionality on a cell phone, a platform with drastically reduced input/output capabilities.

Figure 2 shows the HistView desktop interface. In many cases, the same function can be performed in several views. The views are linked, so the results of an action in one are reflected immediately in all views.

1. The Selection Table offers a tabular interface to a set of categories. The table may be sorted by any column in ascending or descending order. Users can include and exclude categories, set category weights, and view/edit the subcategories of a category. When a category is added to the specification, its initial weight is based on how often it occurred in the history.
2. The Histogram gives a graphical overview of the categories currently in the specification. Users can set weights, remove categories, and view/edit subcategories. For each category, the left bar represents occurrences of the category in the history[1], and the right bar is the *control shadow.* The control shadow represents and lets users modify the category's current weight. Figure 1 shows that, as a user modifies the weight of one category, the system adjusts the weights of other categories proportionally. When the user finishes adjusting a bar (indicated by releasing the mouse button), the system will move it (if necessary) to maintain the categories sorted from largest to smallest weight.
3. The Sample Playlist view shows sample history events consistent with the specification. It lets users define filters and replay events (for this dataset, replaying events means listening to songs). A playlist is a concrete interpretation of a specification that helps users decide whether a specification captures their intent. The effect is related to query by example in that users can modify their specification based on concrete examples.
4. The Navigation / Category Definition bar shows global state information and lets users redefine the categories being viewed. They can do this by forming categories on a different attribute (e.g., *dj* rather than *artist*). Also, if a user is viewing a set of subcategories (e.g., all the albums by The Apples In Stereo), the navigation bar lets users move back up in the specification hierarchy (e.g., to return to viewing the set of artists).
5. Finally, the Subcategories preview in the Histogram window shows subcategories of a category when a user moves the mouse over it. Users can select which attribute is used to form subcategories; in Figure 2, the *album* attribute is being used.

Cell Phone Implementation

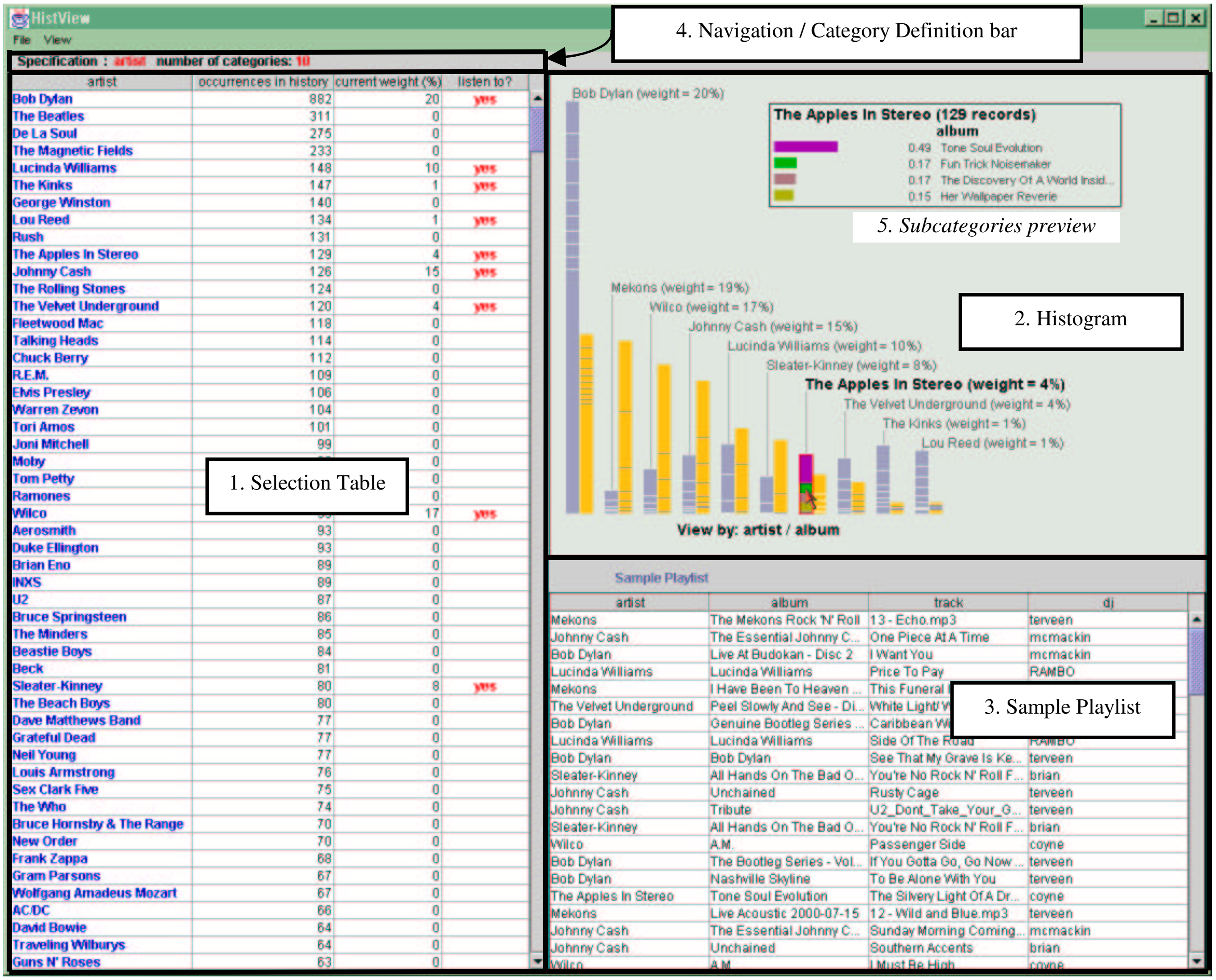
One of our research goals is to explore implementations on mobile devices. Diverse types of applications are being delivered on such devices. Applications include the delivery of streaming audio and personalized news services. Setting preferences is thus as important as for desktop applications, and user history still is a valuable resource.

The constraints of a cell phone platform made it a challenge to implement our system. Indeed, only the newest generation of cell phones makes it possible. We are using one of these devices, the Motorola i85s, as our platform. It runs Java 2 Micro Edition and has a screen with a resolution of 120 x 96. There also is an emulator, which runs on a desktop computer and can be used to test applications before downloading them to the phone. We use screenshots from the emulator to illustrate the cell phone implementation.

Implementing a multi-view, highly interactive graphic application within the constraints of a cell phone platform raises significant design challenges. These constraints, and the specific challenges these posed for us, include:

- Much less screen real estate: from (say) 1024 x 768 to 120 x 96. This made it impossible to display multiple views simultaneously. It also forced us to make tradeoffs between space used for graphic elements (the histogram bars) and text (category labels).
- Fewer colors: the emulator and (apparently) some cell phones support 128 colors; however, the i85s itself provides only 4 shades of gray. This made it harder for us to indicate focus and distinguish subcategories.

[1] Users may choose to hide the history bars; this saves space and is preferable when the occurrence of categories in the history is not important.



Specification : artist number of categories: 10

artist	occurrences in history	current weight (%)	listen to?
Bob Dylan	882	20	yes
The Beatles	311	0	
De La Soul	275	0	
The Magnetic Fields	233	0	
Lucinda Williams	148	10	yes
The Kinks	147	1	yes
George Winston	140	0	
Lou Reed	134	1	yes
Rush	131	0	
The Apples In Stereo	129	4	yes
Johnny Cash	126	15	yes
The Rolling Stones	124	0	
The Velvet Underground	120	4	yes
Fleetwood Mac	118	0	
Talking Heads	114	0	
Chuck Berry	112	0	
R.E.M.	109	0	
Elvis Presley	106	0	
Warren Zevon	104	0	
Tori Amos	101	0	
Joni Mitchell	99	0	
Moby	[illegible]	0	
Tom Petty	[illegible]	0	
Ramones	[illegible]	0	
Wilco	[illegible]	17	yes
Aerosmith	93	0	
Duke Ellington	93	0	
Brian Eno	89	0	
INXS	89	0	
U2	87	0	
Bruce Springsteen	86	0	
The Minders	85	0	
Beastie Boys	84	0	
Beck	81	0	
Sleater-Kinney	80	8	yes
The Beach Boys	80	0	
Dave Matthews Band	77	0	
Grateful Dead	77	0	
Neil Young	77	0	
Louis Armstrong	76	0	
Sex Clark Five	75	0	
The Who	74	0	
Bruce Hornsby & The Range	70	0	
New Order	70	0	
Frank Zappa	68	0	
Gram Parsons	67	0	
Wolfgang Amadeus Mozart	67	0	
AC/DC	66	0	
David Bowie	64	0	
Traveling Wilburys	64	0	
Guns N' Roses	63	0	

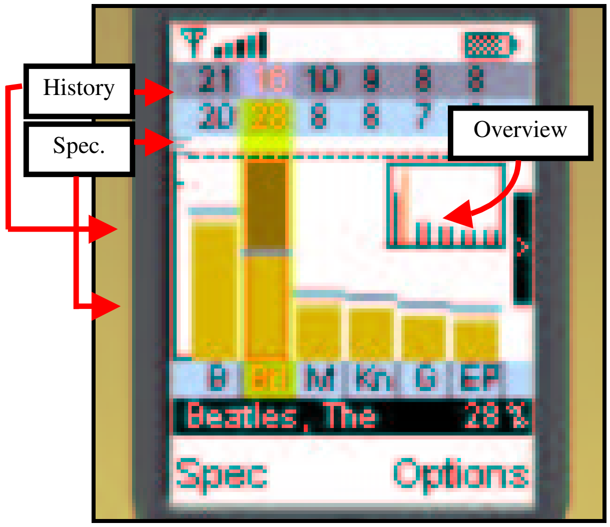
The Apples In Stereo (129 records)

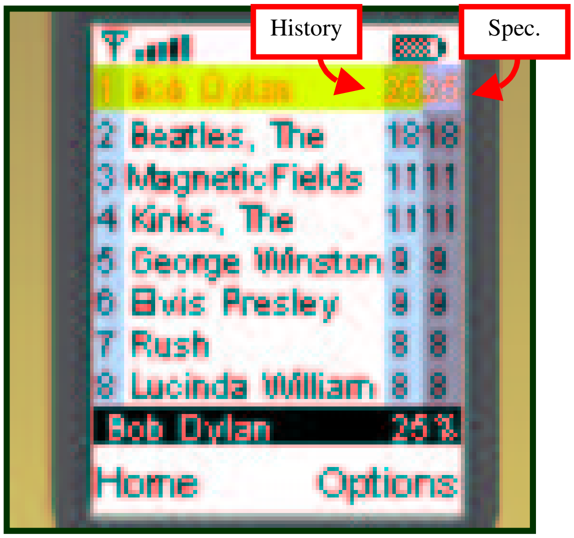
	album	
🟪	0.49	Tone Soul Evolution
🟩	0.17	Fun Trick Noisemaker
🟫	0.17	The Discovery Of A World Insid...
🟨	0.15	Her Wallpaper Reverie

Sample Playlist

artist	album	track	dj
Mekons	The Mekons Rock 'N' Roll	13 - Echo.mp3	terveen
Johnny Cash	The Essential Johnny C...	One Piece At A Time	mcmackin
Bob Dylan	Live At Budokan - Disc 2	I Want You	mcmackin
Lucinda Williams	Lucinda Williams	Price To Pay	RAMBO
Mekons	I Have Been To Heaven ...	This Funeral	[illegible]
The Velvet Underground	Peel Slowly And See - Di...	White Light/ W	[illegible]
Bob Dylan	Genuine Bootleg Series ...	Caribbean Wi	[illegible]
Lucinda Williams	Lucinda Williams	Side Of The Road	RAMBO
Bob Dylan	Bob Dylan	See That My Grave Is Ke...	terveen
Sleater-Kinney	All Hands On The Bad O...	You're No Rock N' Roll F...	brian
Johnny Cash	Unchained	Rusty Cage	terveen
Johnny Cash	Tribute	U2_Dont_Take_Your_G...	terveen
Sleater-Kinney	All Hands On The Bad O...	You're No Rock N' Roll F...	brian
Wilco	A.M.	Passenger Side	coyne
Bob Dylan	The Bootleg Series - Vol...	If You Gotta Go, Go Now ...	terveen
Bob Dylan	Nashville Skyline	To Be Alone With You	terveen
The Apples In Stereo	Tone Soul Evolution	The Silvery Light Of A Dr...	coyne
Mekons	Live Acoustic 2000-07-15	12 - Wild and Blue.mp3	terveen
Johnny Cash	The Essential Johnny C...	Sunday Morning Coming...	mcmackin
Johnny Cash	Unchained	Southern Accents	brian
Wilco	A.M.	I Must Be High	coyne

Figure 2: HistView desktop interface





1	Bob Dylan	26	26
2	Beatles, The	18	18
3	Magnetic Fields	11	11
4	Kinks, The	11	11
5	George Winston	9	9
6	Elvis Presley	9	9
7	Rush	8	8
8	Lucinda William	8	8

Bob Dylan 26 %

Home Options

Figure 3: Cell phone – histogram interface

Figure 4: Cell phone – table interface

- No mouse for input: we had to use the keypad instead.
- The Micro Edition of Java is a minimal implementation of the language; thus we had to re-implement significant parts of our system, including much display code.
- The processor is much less powerful, leading to noticeable delays in response to user actions.

Faced with these challenges, our first goal was to preserve the core functionality of the system. We also wanted to try to implement the control shadows graphical technique. We realized, however, that the advantages of a graphical over a textual interface might disappear on a device like a cell phone, so we carefully designed a text-table interface with identical functionality. Finally, instead of multiple, linked, simultaneously active views, each view would have to stand on its own, and users would move between views by pressing buttons and making menu selections.

Figures 3 and 4 show the histogram and text-table interfaces we developed for the cell phone. (There is both a color and 4-shades-of-gray version – we used the color version for the emulator, but the gray scale version on the cell phone itself and thus in the experiment.) Figure 3 shows that we were able to implement the histogram interface on this platform, but with significant design revisions:

- To save space, we show only one bar per category, the control shadow. It is most useful, since it both represents and gives users control over the current weight. We depict the occurrences of a category in the history as a horizontal line associated with the bar.
- There is not enough space to show the labels for all the categories. Thus, only brief abbreviations of the labels are shown under the bars, with the full name of the focus category shown at the bottom of the screen.
- Bar width is computed automatically, based on the amount of available space; even so, there usually isn't enough room to display all the bars at once. We thus implemented a scrolling mechanism and an overview display (indicated in Figure 3) that shows all the categories and highlights the focus category.
- The left and right arrow keys are used to shift the focus from one bar to the next. The up and down arrow keys are used to increment or decrement the weight of the focus category.
- Finally, two rows at the top show the history and current weight for each category.

Figure 4 illustrates a text table interface with similar functionality. Compared to the graphical interface, the table maximizes the amount of space devoted to textual information, in particular, category names. In addition, each row in the table shows a category's order in the list, its history value, and its current weight. The up and down arrow keys are used to shift focus from one row to the next. The right arrow key is used to increment the weight of the focus category, and the left arrow is used to decrement it.

EXPERIMENT

To review, the main ideas of our approach are (1) user history is a useful resource for specifying preferences, and (2) the control shadows graphical technique is an effective means to specify preferences (specifically, to set relative importance of categories). We wanted to test these ideas empirically. Further, we wanted to investigate the differences between the implementations on the desktop and cell phone; in particular, we wanted to test whether any advantages of the graphical technique held for both.

Design

We used the shared music history dataset for the experiment. 24 subjects participated. Subjects included members of our laboratory and summer students; ages ranged from early 20s to early 50s. All were heavy computer users, but most were light-to-medium cellphone users. The first set of experimental tasks was done using the desktop interface, and the second set of tasks was done on the cell phone.

Desktop

The first phase of the experiment used a within-subjects design. Subjects performed a series of 6 tasks consisting of setting the weights of a specified subset of a given set of artists. The purpose of these tasks was to test the histogram interface against a text-table interface with identical functionality. Half the subjects used the histogram interface for the first 3 tasks and the table for the next 3 tasks, while the order was reversed for the rest of the subjects. Before each set of 3 tasks, the experimenter demonstrated how to use the interface, and the subjects did a simple practice task.

Subjects used a modified version of the HistView interface (Figure 2) for the experiment. The Sample Playlist view was not shown at all, and the Selection Table and Histogram views were of equal size. When the subjects were using the table interface, the other view was blank, and vice versa. This meant that subjects had to do each task using solely the functionalities of one of these views.

The second phase of the experiment used a between-subjects design. Subjects were shown how to use the table to select artists and then instructed to select 20 artists they wanted to listen to themselves. The purpose of this task was to test the utility of history data in finding artists to listen to. There were three conditions for the selection task:

- *Personal history* – 8 of the subjects had access to their personal listening histories. This meant that the "occurrences in history" column of the Selection Table showed the number of times they listened to an artist, and the table was ordered by this column.
- *Group history* – 8 of the subjects (not users of the listening environment) had access to the overall group listening history. The "occurrences in history" column showed how often anyone listened to an artist, and the table again was ordered by this column.

- *No history* – 8 of the subjects were not given access to any history information. The "occurrences in history" column was not shown at all, and the artists were ordered alphabetically.

Subjects were not allowed to re-sort the table, thus forcing them to scan artists in the order defined by the history condition they were assigned to.

The third phase of the experiment again used a within-subjects design. The goal again was to test the relative utility of the histogram interface, this time on a more realistic task, with data that was meaningful to the subject. To do this, the set of 20 artists selected by a subject was divided into two. Each subject was asked to weight the artists in first one subset, then the other, according to their personal preferences. Half of the subjects used the histogram for the first subset of artists and the table for the second subset, while the order was reversed for the rest of the subjects.

After each phase of the experiment, subjects were asked a few questions about how well the interfaces they used had supported the task and (in the first and third phases) whether they preferred one interface over the other.

Cell phone

Subjects performed one weighting task using the histogram interface (Figure 3) and one using the text table (Figure 4). The design was within-subjects. Subjects who used the histogram interface first on the desktop also used it first on the cellphone, and vice versa. Again, subjects were instructed which artists' weights to adjust and how, and subjects practiced each interface before performing the assigned task. After the tasks were completed, subjects were presented evaluation questions about the two interfaces just as in the desktop condition.

Results

1. Histogram desktop interface more effective and preferred

On the first set of weighting tasks, subjects took less time with the histogram interface than with the text table interface. On average, across the three trials, subjects using the histogram took 49.5 seconds while subjects using the table took 58.6 seconds (One-Factor ANOVA, p<0.0005). When asked to evaluate the interfaces, subjects:

- rated the histogram interface easier to use – 4.6 on a 5 point scale vs. 3.3 for the text table, (Paired Samples T-Test, p<0.0003), and
- strongly preferred the histogram interface – subjects disagreed with the statement "For setting relative importance, I preferred the table interface to the histogram interface" (1.9 on a 1-5 scale).

The second weighting task was more realistic since subjects were dealing with artists that they had selected. Once again, subjects completed their tasks more quickly using the histogram (histogram: 83.6 seconds, text table: 100.1 seconds; One-Factor ANOVA, p<0.003). We also measured the amount of work performed in each condition by counting the number of times subjects adjusted the weight of any artist. Subjects using the histogram interface actually performed more work (9.1 weight change operations) than subjects using the text table (6.9 operations; One-Factor ANOVA, p<0.02). Table 1 summarizes these results.

Table 1: Second weighting task

	Time	***Weight Change Operations***	***Seconds per Operation***
Histogram	83.6s	9.1	9.2
Text Table	100.1s	6.9	14.5

Questions similar to the ones asked after the first weighting task were administered after the second weighting task. Subjects agreed even more strongly that the histogram interface was preferable to the text table (4.5/5), and they continued to rate the histogram interface as significantly easier to use than the table (4.5/5 vs. 2.8/5; Paired Samples T-Test, p<0.0001). Subjects may have increased their preference for the histogram because they were working with artists they had chosen themselves and because they did more work in each interface.

Subject comments support and help explain these results. First, a common observation was that the histogram gave an effective overview of a set of categories and their relationships – it was "easy to understand the relative importance of the categories with a quick look". Second, subjects liked the dynamic feedback of the histogram interface; that is, as they stretched a bar, they found it useful to see the other bars change in response. This made it easy to change weights, and, in particular, to see what the global effects of a change would be – it was "easy to visualize the new state that was being created", "very easy to eyeball the relative effectives of my changes as I made them", and "possible to very quickly see the results of one's potential actions".

A common complaint about the table was that subjects did not like selecting a specific numeric weight: "I prefer just seeing relative weights". In contrast to the histogram, it was "difficult to see relative importance" of categories. Further, subjects missed visual feedback that showed how changing the weight of one category would effect the others – "it was a bit confusing to type in numbers not knowing how it would affect other values; the histogram allowed me to visualize this" and it was "impossible to gauge the effects on other values of making complex changes".

2. Personal history was useful; group history was not

We examined the effects that history (personal, group, or none) had on subjects' selection and weighting tasks. We first looked at the time it took for each subject to select their favorite 20 artists. As expected, subjects in the personal history condition performed this task most quickly (199 seconds). We also expected that group history actually would hinder performance unless it happened to match a subject's personal preferences, and the numbers supported this expectation – group history subjects took 296 seconds while subjects in the no history condition took 215 seconds.

Other analyses, however, did not show an advantage for personal history subjects. An examination of their actual listening history revealed a likely explanation: 4 of these subjects were heavy users of the environment, having played between 775 and 1940 tracks, while the other 4 were light users, playing between 108 and 221 tracks. The first 4 subjects thus had extensive history, which represented their preferences accurately, while the histories of the other 4 did not represent their preferences nearly as well.

This supposition was supported when we reanalyzed the time data for the selection task. The 4 "heavy users" made their selections in an average of 117 seconds; however, the 4 "light users" average 280 seconds, performing at about the same level as group history subjects. The time taken by personal history "heavy users" was significantly less than that taken by subjects in the other 3 conditions (Independent Samples T-Test, p = 0.05).

In the next analysis, we examined the amount of editing subjects in each history condition did to the sets of artists they selected. We continued to treat the two types of personal history subjects separately. Consistent with the previous analysis, the personal history "heavy users" made fewest changes – this makes sense since the weights of the artists they selected were based on large amounts of listening history, so should reflect their tastes well – the "no history" subjects were next, and the personal history "light users" and the group history subjects made the most changes. Table 2 summarizes the results.

Table 2: Weight Change Operations by History Condition

	Histogram	Text Table	Average
Personal/heavy users	6.8	4.8	5.8
Personal/light users	7.8	9.8	8.8
Group History	11.0	8.5	9.8
No History	8.9	5.1	7.0

Subject comments bore out these observations. Several personal history "heavy users" remarked how useful it was to have their favorite artists at the top of the list: "I just want to drag-and-drop the first 20 artists". A personal history "light user" commented that "it was easy to selected from the top", i.e., from the small set of artists for which he had listening history, but it was hard thereafter. On the other hand, several group history subjects volunteered that the listening history was completely useless. Since it did not reflect their own preferences, the order of artists seemed random to these subjects.

3. No advantage for histogram interface on the cell phone

On the cell phone, subjects did complete their task faster using the histogram interface than the text table interface (histogram: 126sec; table: 134sec); however, the difference was not significant (One-Factor ANOVA, p=0.11).

The evaluation questions showed reduced user satisfaction with both interfaces – 3.4/5 for the histogram and 3.2/5 for the table. Further, the user preference for the histogram disappeared: subjects were nearly neutral with respect to the statement "For setting relative importance, I preferred the table interface to the histogram interface" (2.5/5).

Subject comments showed why this was the case. Several subjects reported difficulties in seeing the graphics and text; these comments applied to both interfaces. The key problem, however, was speed – both the histogram and table were slow on the cellphone, but the histogram was slower due to time needed to redisplay the graphics: "the cellphone [histogram] interface was so slow that it was difficult to make judgments". Second, stretching the bars using the up and down arrow keys was not as direct as stretching with a mouse: "pushing buttons [was] not as easy as dragging with a mouse".

On the other hand, the text table was relatively better on the cell phone. First, it was faster: "response time was better than the histogram". Second, it was beneficial to be able to "see the full name of the artist".

4. User experience of the various interfaces

At the end of the experiment, we asked subjects a series of questions taken from the QUIS instrument [3] to gauge their subjective experience. Subjects rated each of the four interface/platform conditions (on a scale of 0-6) on the following three dimensions: terrible to wonderful, hard to easy, and frustrating to satisfying.

Table 3: QUIS question results

	Terrible/ Wonderful	*Hard/ Easy*	*Frustrating/Satisfying*
Histogram/Desktop	5.17	5.30	4.96
Histogram/CellPhone	3.74	3.57	3.0
Table/Desktop	3.35	3.17	2.96
Table/CellPhone	3.04	3.43	2.83

Table 3 shows the results, which are consistent with previous subjective preferences and objective measures. On the desktop, subjects significantly preferred the histogram to the text table (Paired Samples T-Test, p < 0.001). However, the histogram fared much worse on the cell phone, with its advantage over the text table virtually disappearing (the differences were not significant).

User comments and implications for redesign

Subject comments revealed a number of issues. First, while most subjects liked the dynamic feedback of the histogram interface, some did not. Instead, they wanted to get all the category weights just right themselves, then explicitly instruct the system to adjust all weights to add up to 100%. A related notion also came up: some subjects wanted to be able to "lock" categories, that is, to prevent their weights from being adjusted automatically. The idea is that once they "got an artist right" (say, set its weight to 20%), they didn't want it to be changed unless they said it should be.

There were several smaller issues regarding the histogram. A few subjects wanted to see a category's numeric weight updated immediately as they were stretching a bar. They seemed to have the goal of setting a weight to (say) 25%, and it was not easy to do this. Some subjects found the history bars useless; however, others liked having them as a reference. As we mentioned, the display of history bars can be turned off; however, to keep the experimental tasks simple, we did not instruct subjects about this function.

Second, a number of subjects had a different model of how the table interface (especially on the cell phone) should have worked. Rather than setting the weights, they wanted to manipulate the order of the categories, i.e., move categories up and down in the list. Several subjects also said they'd first like to get the order right, then tweak the weights. (Again, this functionality is implemented, but subjects weren't instructed in its use).

DISCUSSION

This research illustrates lessons to be considered when moving an interface from a desktop computer to a mobile device. First, it is possible to implement a sophisticated graphical technique on current state-of-the-art mobile devices,. However, hardware and software constraints for these platforms play a much larger role than in current desktop computing environments. It is difficult to give a visual overview of large amounts of data or to sustain a feel of direct manipulation using keypad input. Slow processors further hinder the interaction, with desirable redisplay and response times impossible to meet. These considerations argue that simpler interfaces may be better, when they can be designed with the necessary functionality. In our case, the table interface performs about as well as the histogram (and we believe a few added features could make it better).

We also learned some lessons about history data. First, we saw that users' personal history generally was very helpful for reuse tasks. Even here there was a small surprise: a few users, while making effective use of their history, wanted recommendations of music they weren't already familiar with, rather than just being given effective access to music they already knew. Second, providing access to the history of other users is not helpful when their preferences do not match one's own. This observation supports the argument that history reuse must blend into collaborative filtering – users must receive support in finding other users with similar tastes. These users' history is likely to be useful – and since it is not identical, it also may address the request of a few personal history users for novel recommendations.

REFERENCES

[1] Ahlberg, C. and Shneiderman, B. The Alphaslider: A Compact and Rapid Selector, in *Proc. of CHI'94* (Boston MA, April 1994), ACM Press, 365-371.

[2] Ball, T.A. and Eick, S.G. (1996). Software visualization in the large. IEEE Computer, 29(4), 33-43.

[3] Chin, J.P., Diehl, V.A., Norman, K.L. Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface, in *Proc. of CHI'88*. (Washington DC, May 1988), ACM Press, 213-218.

[4] Eick, S. Data Visualization Sliders, in *Proceedings of UIST'94* (Marina del Rey CA, November 1994), ACM Press, 119-120.

[5] Greenberg, S. (1993). The computer user as toolsmith: The use, reuse, and organization of computer-based tools. Cambridge University Press, Cambridge.

[6] Hanson, S.J., Kraut, R.J., and Farber, J.M. Interface Design and Multivariate Analysis of UNIX Command Use. *ACM Transactions on Office Information Systems* 2(1), (1984), 42-57.

[7] Herndon, K., Zeleznik, R., Robbins, D., Conner, D.B., Snibble, S., and van Dam, A. Interactive Shadows, in *Proceedings of UIST'92* (Monterey CA, November 1992), ACM Press, 1-6.

[8] Hill, W.C., Hollan, J.D., Wroblewski, D., and McCandless, T. Edit Wear and Read Wear, in *Proceedings of CHI' 92* (Monterey CA, May 1992), ACM Press, 3-9.

[9] Hill, W.C. and Hollan, J.D. History-Enriched Digital Objects: Prototypes and Policy Issues. *The Information Society* 10(2), (1994), 139-145.

[10] Lee, A. Investigations into History Tools for user Support, Unpublished Ph.D. Dissertation, University of Toronto, 1992.

[11] Maltz, D. and Ehrlich, K. Pointing the Way: Active Collaborative Filtering, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 202-209.

[12] Munro, A.J, Höök, K. and Benyon, D (Eds.) *Social Navigation of Information Space*. Springer, 1999.

[13] Resnick, P. and Varian, H.R., guest editors, *Communications of the ACM,* Special issue on Recommender Systems, 40, 3 (March 1997).

[14] Shneiderman, B. Dynamic Queries for Visual Information Seeking. IEEE Software, 11,6 (1994), 70-77.

[15] Tweedie, L.A., Spence, R., Williams, D., and Bhogal, R. The Attribute Explorer, in *Conference Companion of CHI'94* (Boston MA, April 1994), ACM, 435-436.

[16] Wexelblat, A. and Maes, P. Footprints: History-Rich Tools for Information Foraging, in *Proc. of CHI'99* (Pittsburgh PA, May 1990), ACM Press, 270-277.

[17] Wittenburg, K. Lanning, T., Heinrichs, M., and Stanton, M. Parallel Bargrams for Consumer-based Information Exploration and Choice, in *Proc. of UIST'01* (Orlando FL, November 2001), ACM Press.