The Design and Evaluation of a Task-Centered Battery Interface

Khai N. Truong[1], Julie A. Kientz[2], Timothy Sohn[3], Alyssa Rosenzweig[1], Amanda Fonville[2], Tim Smith[1]

[1]University of Toronto
Toronto, Ontario, Canada
{khai, alyssacr, tsmith}@cs.toronto.edu

[2]University of Washington
Seattle, Washington, USA
{jkientz, ajf32}@uw.edu

[3]Nokia Research Center
Palo Alto, California, USA
tim.sohn@nokia.com

ABSTRACT

Battery interfaces provide important feedback about how much time users can continue using their mobile devices. Based on this information, they may develop mental models of the types of activities, tasks, and applications they can use before needing to recharge. Many of today's battery interfaces tend to report energy in coarse granularities or are highly inaccurate. As a result, users may find it difficult to depend on the estimates given. We conducted a survey with 104 participants to understand how users interact with various mobile battery interfaces. Based on the survey results, we designed and prototyped a task-centered battery interface on a mobile device that shows more accurate information about how long individual and combinations of tasks with several applications can be performed. Our pilot study of eight users demonstrated that fine-grained information separated by tasks can help users be more effective with and increase their understanding of their device's battery usage.

Author Keywords

battery interface, mobile computing, ubiquitous computing

ACM Classification Keywords

H5.2 Information Systems: User Interfaces – User-Centered Design

General Terms

Human Factors, Design

INTRODUCTION

The amount of power remaining can substantially impact how people use applications and ultimately their mobile devices. Many mobile devices include a battery interface that provides high-level feedback about the remaining charge. Users often view these battery interfaces to determine how long they can continue to use the device before recharging. Thus, it is important to present this information in an accurate and easily comprehensible interface.

UbiComp 2010, Sep 26 – Sep 29, 2010, Copenhagen, Denmark.

Many of today's battery interfaces present a simple view of the underlying voltage capacity in the form of an overall percentage or amount of time remaining. The percentage and time remaining are typically created from models and assumptions about the actual battery. These interfaces are more user-friendly than pure voltage numbers, but do not provide the user with a clear idea of how long she can run different applications on her mobile device given the remaining amount of charge on her battery. The user needs to create her own mental model about how the battery discharges and how the remaining battery percentage shown in the interface correlates to different application usage (*e.g.*, web browsing, watching a video using the media player). The user can typically create this model over time using the device, but is always left to guess at whether a task can be completed in the time remaining.

This paper presents the design, implementation, and evaluation of a Task-Centered Battery Interface (TCBI) that provides users with an estimated amount of time remaining for common applications running independently or in combination to eliminate the guesswork that is often involved with traditional battery interfaces. Our goal is not to improve power management or create an adaptive power interface based on user behavior, but rather focus solely on the interface and presentation of the underlying power management.

To aid in the design of the interface, we conducted a survey with 104 participants to understand users' battery usage habits and their perception of existing battery interfaces. TCBI is similar in spirit to several existing battery interfaces, such as EZ Battery Life [8], Battery Time [6] and so on, but goes beyond them by displaying battery time remaining for both single and multiple applications. We conducted a four-week pilot study with eight users to evaluate how effective and informative TCBI would be in real-world use. Participants were given a Nokia N810 Internet tablet to use for four weeks in total; two weeks with the standard battery interface and two weeks with TCBI complementing the standard interface. We found that all participants preferred the TCBI interface compared to the standard percentage/time remaining interface. Our results demonstrate that participants were able to determine easily and quickly whether they had enough battery life to perform activities on their mobile device.

RELATED WORK

Energy conservation on mobile devices has been a widely studied area of research. Much of the related work has been at the system level of power management, creating energy-efficient programs, and improving human-battery interaction. The Turducken system, for example, offers a hierarchical power management tool for mobile devices that utilizes multiple devices to improve battery life [17]. Researchers have also explored using the characteristics of different wireless radios to improve power management [14] and self-tuning power management based on the access patterns of applications [1]. Other projects have explored using energy as another system resource that is managed within the operating system based on application needs [19]. Flinn and Satyanarayanan showed that a collaborative relationship between applications and the operating system can help extend battery duration for user goals [9].

In addition to optimizing system resources to preserve energy, other techniques have looked at how to improve power usage in interactive applications. Researchers at HP Labs created an energy-adaptive display based on OLED technology that selectively darkens parts of the screen based on salient areas [11]. User acceptance of the interface was shown to be acceptable in a two-part study [7, 10]. Vallerio *et al.* proposed several techniques for creating energy-efficient GUIs that could improve overall system energy [18]. Recently, researchers have created energy-efficient parallel software for mobile devices as a means to conserve battery life [12]. Ravi *et al.* proposed a context-aware battery management system that would predict how long the current set of applications would continue to execute [16]. The system would warn the user if it detected that the mobile phone battery may run out before the next recharging opportunity. These approaches are useful for creating a system that manages power usage without wasting unnecessary resources and are complementary to our work on studying the battery interface.

Rahmati *et al.* found that existing battery indicators are inadequate and the user would benefit from a higher resolution, more informative battery interface [15]. Our work builds on their insights by evaluating a task-centered battery interface to provide the user with more accurate information. Banerjee *et al.* also studied the user's interaction with batteries, but focused mainly on recharging habits [2]. Based on their findings, they proposed Llama, a user-adaptive energy management system that harvests excess battery energy to create a better user experience. Llama focuses on understanding the user's charging habits and not improving the actual interface presented to the user. A user-adaptive power system could help improve a task-centered battery interface.

There are several commercial smartphone applications, such as EZ Battery Life [8], Battery Time [6], BatteryGo [3], myBattery [13], Battery Juice [4], Battery Magic [5] and so on, that show time remaining estimates for different tasks such as talk time, Internet time, and audio/video playback. There is no public report on how the estimates are given, but we found these battery interfaces to be inaccurate by up to 1 hour during our lab studies in which we used different supported applications (such as video playback) on the smartphones for a predetermined amount of time. Moreover, the interface displays time estimates assuming only one application will be active at a time. However, in real practice, a user often may perform multiple tasks at once (*e.g.*, listening to music while browsing the Web). Our work is in the spirit of these interfaces, but takes them a step further by offering time estimates for simultaneous tasks and also provides an evaluation of the underlying energy model. In addition, there is no reported study on the comprehensibility of these interfaces compared to more traditional interfaces, which our research aims to explore.

SURVEY OF BATTERY USAGE & PERCEPTIONS

We conducted a wide-scale online survey to better understand people's current battery usage on mobile devices. The questionnaire consisted of 26 questions – 20 closed-ended and 6 open-ended. The questions asked about demographics, different portable devices used, use of the battery interface and its perceived accuracy and usefulness, and the frequency and impact on behavior of a low-battery warning and loss of power before being able to recharge the battery or in the middle of usage. We recruited participants through a variety of email mailing lists and online advertisements. In total, 104 completed the survey (52 males and 52 females). Participants ranged in profession, including an engineer, environmentalist, chiropractor, TV editor, and fitness trainer. The age of respondents varied: 18-19 (4), 20-29 (36), 30-39 (35), 40-49 (20), 50-59 (7) and 60+ (2). Of the study participants, 85% considered themselves advanced or expert computer users. All participants were entered into a drawing to win one of four $25 gift certificates to an online bookstore.

Device Usage

Participants indicated their device ownership and usage for all mobile devices, with 95% owning a smartphone/cell phone/PDA and 40% of those participants owning 2 or more of these devices. Similarly, 93% owned a laptop and 43% of those participants owned two or more. Of the respondents, 72% owned an Internet tablet or media player, and 7% reported owning another type of mobile device (*e.g.*, Amazon Kindle). In total, the average participant owned 4.26 devices (SD: 2.22). Most smartphone and laptop users used their device on a daily basis, with 6% of smartphone owners and 8% of laptop owners indicating they use their device only on a weekly/monthly basis. Internet tablet/media player users varied widely in the frequency of their device usage, with only 57% of Internet tablet/media player users using their device on a daily basis.

Battery Death

We asked participants to estimate how frequently their mobile devices die both while idle and in use. Smartphones/cell phones died more frequently while idle than any of the other mobile devices. Twenty-one

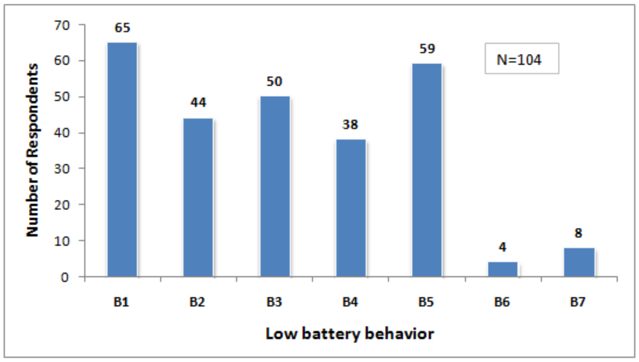

Figure 1: Breakdown of responses for low-battery behavior when user does not have access to an external power source.

participants reported their smartphone/cell phone dies at least once per week, and 49 participants reporting it dying once or more per month. Laptops had the second most reported battery deaths[1] followed by Internet tablets. Battery deaths were slightly less frequent while the device was in use, and again smartphones/cell phones were the most common, with 16 participants reporting losing all battery power while using the device at least once per week, and 29 indicating at least once per month. Following smartphone/cell phones, Internet tablets were reported to die while in use more frequently than laptops on a weekly basis, but laptops died more frequently than Internet tablets on a monthly basis while in use.

Low Battery Behavior

Each participant rated the frequency of low-battery warnings on a weekly scale for each device. Of the smartphone users, 63% indicated they see a low-battery warning at least 1-2 times per week with 18% of those seeing one between 3-9 times per week. Laptop users also had a similar experience with 53% receiving a low-battery warning 1-2 times a week. We asked participants to describe their behavior when a low-battery warning is received on any of their portable devices and they do not have access to an external power source. They were given seven choices and asked to select all of the options that best describe their behavior.

- ***B1:*** *I limit my use to emergency/absolutely necessary situations*
- ***B2:*** *I adjust device settings (*e.g.*, brightness, display time) to conserve power*
- ***B3:*** *I change which applications I am using (*e.g.*, only use text messaging, stop Internet use, or phone conversations)*
- ***B4:*** *I turn off wireless capabilities*
- ***B5:*** *I take measures to preserve my current activities and/or saved documents*
- ***B6:*** *I do not take actions to conserve power and I continue to use it as if I did not receive the message*
- ***B7:*** *Other behavior*

[1] Throughout this paper, we use the colloquial expressions of 'death' and 'dying' to describe the state in which the battery is so limited that it can no longer power the device.

Cell phone users often may need to make or receive important phone calls. This common behavior was confirmed by substantial responses (65) indicating that respondents limit device use for "emergency situations" (see Figure 1). On laptops, however, documents and other work might need to be saved prior to batteries "dying" and data being lost, behavior indicated by the second most popular response (see Figure 1). Expert users seemed to be more likely to change device settings or turn off wireless capabilities, whereas intermediate users were more likely to change application behavior, such as stopping phone conversations or Internet usage, rather than changing hardware settings (*e.g.*, wireless radio). However, we did not have enough novice or intermediate participants to draw statistically significant conclusions.

A Better Battery Interface

Our survey results indicate the most important factors related to battery interfaces are accuracy of time remaining and ample warning about an impending battery death. Participants tended to create their own models about the relationship between actual time remaining and the information reported on the battery interface. We asked participants questions that gave us an indication of how they create mental models of their battery interfaces. Some were based on past experience or correlations of the expected battery life of the product from marketing materials. The general sentiment was that the reported times and interface were expected to be linear, but did not often behave accordingly. Two participants noted:

> *"Usually, my cell phone battery shows that it is fully charged - as soon as it drops down one bar, all the other bars go quickly." **–P52***

> *"The laptop's battery indicator is so unreliable. It says something like 30% remaining and then it starts warning 3% level at the next moment." **–P58***

We found that battery interfaces could also be improved by showing time remaining for different applications. Figure 2 shows the agreement levels from participants with "*I think the current battery interface on my portable device would be more useful if it showed information that relates specifically to the application I am using.*" The average

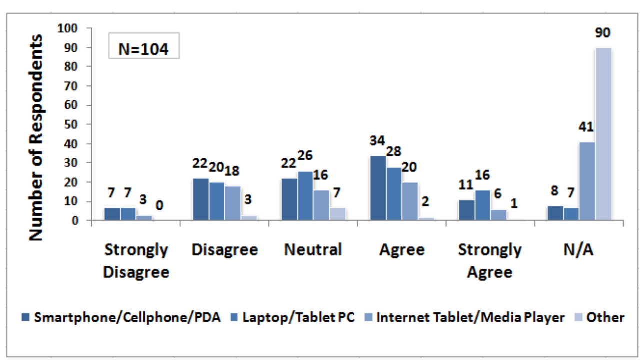

Figure 2: Participant agreement levels when asked "I think the current battery interface on my portable device would be more useful if it showed information that relates specifically to the application I am using."

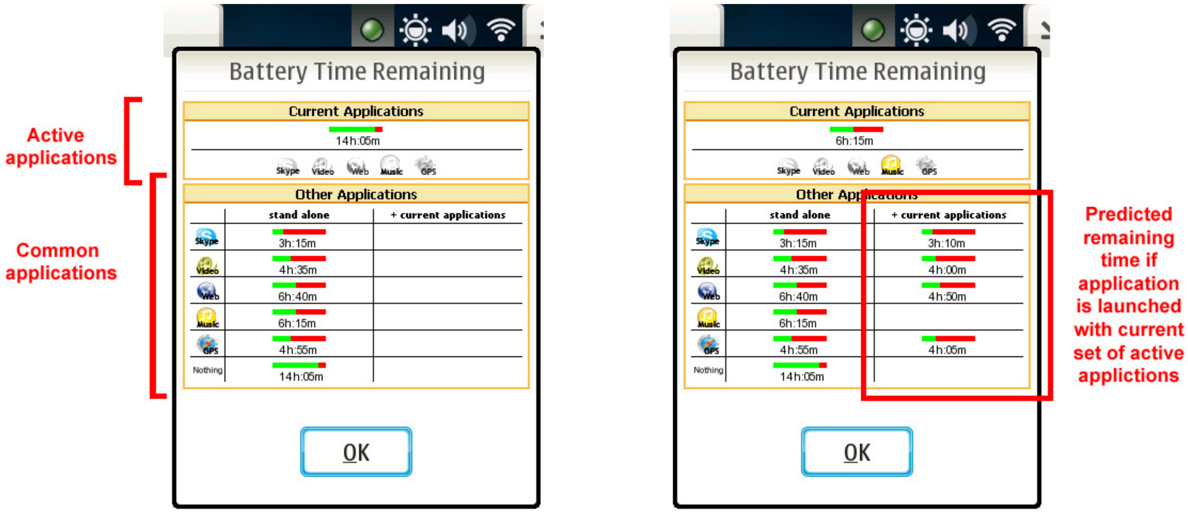

Figure 3: The Task-Centered Battery Interface. The left figure (a) shows the battery time remaining when none of the common applications is running. The right figure (b) shows the battery time remaining when the user is listening to music and estimates.

response tended to be more neutral. Some participants only wanted the big picture and did not feel this extra information would help.

> *"I know generally what applications are running and I know that the fewer I run the longer my battery will last so I know that shutting some down will save power. So knowing that certain applications are eating power is not really helpful"* ***–SP18***

Others felt that having this applications-level information could help them make better decisions about which applications to keep open and which to shutdown. They saw this interface as complementary to the big-picture interface.

> *"I still want to have this general battery life indication but it would be useful to see easily which applications are really eating up the power when I need to constrain the power usage."* ***–SP15***

> *"It would be great if cell phones could have some sort of indicator…. Like given the current power level of your battery and what applications/programs you have open, you have this much time left before your battery is drained if you only used this one application/program. It could even be a list of using x application would leave you with this much time to actually be in it, given your current level of battery."* ***–SP63***

Users could benefit from a battery interface that complements existing interfaces. Specifically, of the respondents for laptops, 44 indicated they would benefit from an application-specific battery interface, 27 said they would not benefit, 26 remained neutral, and 7 did not respond (see Figure 2). In the following sections, we describe the design, implementation, and initial evaluation for our task-centered battery interface that shows application-specific information for time remaining.

TASK-CENTERED BATTERY INTERFACE

Based on the survey results, we developed a Task-Centered Battery Interface (TCBI) that provides users with estimates for the amount of time available for common tasks on mobile devices. In this section, we describe the user interface and implementation details for TCBI on the Nokia N810[2], a portable, Wi-Fi enabled Internet tablet with no cell phone capabilities.

High-Level Design

TCBI presents a breakdown and time remaining for individual application usage to eliminate the guesswork that is often involved with traditional battery interfaces. Unlike existing battery interfaces, TCBI estimates how much battery time remains given the current set of active applications (see top of Figure 3a). In addition, the interface allows the user to make predictions on how launching a common application will impact the remaining battery time (see bottom of Figure 3b). TCBI displays how long the user can expect to use an application by itself or when it is used along with the current set of active applications.

Implementation Details

We developed TCBI as a Python status bar applet that is shown when the user clicks on its icon. When launched, the applet obtains 1) the battery's current voltage reading and 2) a list of the processes running. TCBI looks up the battery's current voltage value in a battery usage profile lookup table to obtain the battery time remaining for the individual applications. It uses the battery time remaining for the individual applications to estimate the battery time remaining for the active set of applications, as well as potential combinations if any of the common applications is launched next.

Battery Usage Profile Lookup Table

To determine how long different applications can continue to be used on the device, we developed logging software for the Nokia N810 to record time-stamped battery readings every 10 seconds while an application runs alone in the

[2] When we first developed the interface, the Nokia N900 had not yet been released. This application should be generalizable to that new device.

foreground. We used this logging software to gather battery traces of the following applications used in these ways:

- **None:** the device is left on with no application in use;
- **Video player:** a list of movies is played in a continuous loop full screened;
- **Music player:** a song list is played in a continuous loop;
- **Skype:** the device is connected with another N810, each with an audio source playing in the background;
- **GPS:** the map application is running with GPS tracking enabled;
- **Web browser:** a user browses the Web continuously.

We gathered six traces for each application. For all traces, we set both the device's screen brightness and audio volume at 50%, and the display always remained on. We used these settings along with brand new batteries to maintain consistency among all gathered traces. Batteries do wear out with time as they are drained and recharged, but we did not collect traces from heavily used batteries because the focus of our work is primarily on the interface issues.

For each set of application traces, we averaged the logged time remaining for each voltage capacity reading. We then performed a smoothing function on the resulting battery usage profile for each application. We cross-validated each set of application traces against their respective battery usage profile. Overall, when only one application is being used, this approach produces predicted battery time remaining values that come within an average of ~738.37 seconds of the actual battery time remaining or ~12.7% of the actual time remaining. Thus, if an hour of battery time is remaining, the prediction would be off by ~7.62 minutes on average; and when there is 15 minutes remaining, the prediction would be only off by ~2 minutes on average. Actual time remaining was measured by completely depleting the battery. The average error of individual battery usage profiles is reported in Table 1.

We then constructed a lookup table containing the battery usage profile for the 5 applications listed above. The lookup table provides a mapping from voltage to time remaining for each application. This battery usage profile lookup table allows the system to quickly identify the expected battery time remaining if any of the applications are being used or none at all. Although we currently only predict battery time remaining for 5 applications, the described approach is generalizable for including additional applications.

Table 1: The average error for each application's battery usage profile. Magnitude error is the average difference between the predicted battery time remaining with the actual battery time remaining. Percentage error is the average percentage difference between the predicted battery time remaining with the actual battery time remaining.

Application	Magnitude Error	Percentage Error
GPS	615.88 sec	13.0 %
Music	859.55 sec	14.4 %
Skype	685.77 sec	26.7 %
Video	649.25 sec	13.8 %
Web	1400.41 sec	21.6 %
Nothing	1696.12 sec	11.9 %

Table 2: The average error for our multi-application battery time remaining estimation function.

Application	Magnitude Error	Percentage Error
GPS + Music	388.30 sec	9.4 %
GPS + Skype	500.21 sec	18.1 %
GPS + Video	378.08 sec	8.9 %
Web + Music	981.17 sec	21.9%
Web + Skype	895.76 sec	21.4%
Web + GPS +Music	1472.42 sec	23.1%

Multi-application Battery Time Remaining Estimation

We developed a function to estimate battery time remaining when multiple applications are used simultaneously. Although each application is inherently different, the same resources (*e.g.*, screen, memory) will be jointly used. Thus, battery time remaining is computed by assuming that each application will drain the battery in the following manner:

$$\mathbf{t}_{predicted} = \boldsymbol{C} \times 1 \,/\, (\, 1/\mathbf{t}_1 + \cdots + 1/\mathbf{t}_n), \text{ where}$$
$$\boldsymbol{C} = (\boldsymbol{n} + 1) \,/\, \boldsymbol{n}, \text{ and}$$
$$\boldsymbol{n} = \text{the number of applications running.}$$

To validate this estimation function, we gathered traces for 6 different application combinations. For each combination, we produced a predicted battery usage profile. We then again cross-validated each set of traces against their respective

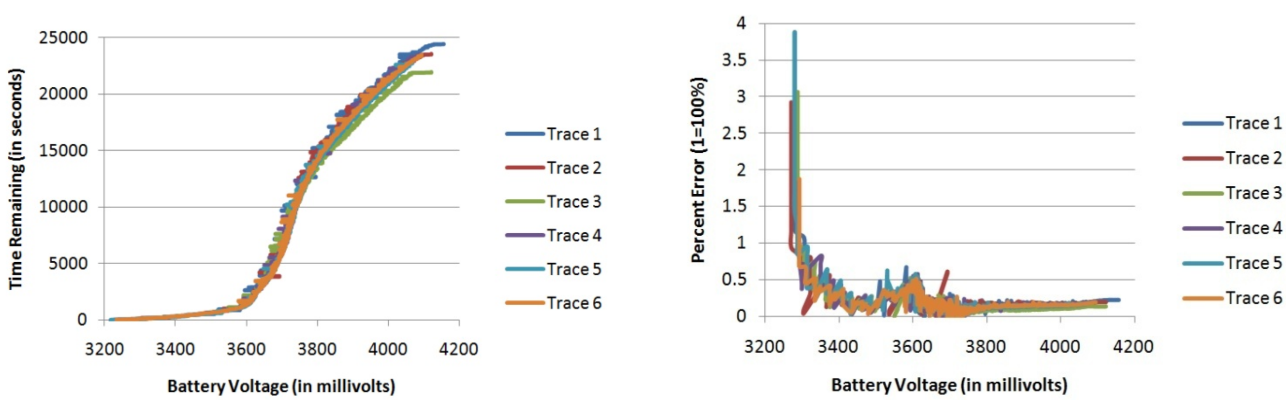

Figure 4: Graphs showing video playback battery traces (left) and the percent error in the predicted value in relation to the actual time remaining (right).

battery usage profile. The predicted battery time remaining values come within an average of ~769.3 seconds of actual battery time remaining or ~17.1% of actual time remaining. Table 2 reports the average error for the multi-application battery time remaining estimation function.

DEPLOYMENT STUDY

To understand how TCBI improves user understanding of task-time remaining, we conducted a four-week user study comparing the new interface with the default interface on the Nokia N810. In this section, we describe the study method, participant recruitment, and results.

Study Method

The main purpose of the user study was to examine the effect of TCBI in comparison with the Nokia N810's default battery interface on users' perceptions of how different applications affect battery usage. Thus, we conducted a two-phase, within-subjects deployment study over four weeks with eight participants. For the first two-week phase of the study, participants used either the TCBI interface or the standard Nokia battery interface and switched to the other during the second two-week phase.

To ensure the Nokia device was used throughout the study, we gave each participant a set of 5 tasks to be conducted on 5 different days over the two-week period of each phase (for a total of 10 tasks). These tasks involved using at least one of the profiled applications for approximately 20 to 30 minutes. Example tasks included calling airline customer service numbers with Skype to find flight times, listening to music and writing down words from the lyrics, watching 30 minutes worth of movie trailers and describing the movie topics, and visiting news websites and writing down the authors of articles. Although participants could also use the devices for more practical activities, we chose to have them also complete specified tasks to ensure that participants all had a similar baseline experience with the device when answering our questions about the battery time remaining. We intentionally did not design the study to require use beyond the scheduled tasks because we wanted to understand how the two interfaces differed in their impact on the participants' mental models across different adoption and usage behaviors. The within-subject design of the study ensures that the same adoption and usage behaviors are consistent across both phases of the study.

The tasks for the two phases were similar in nature, but resulted different outputs (*e.g.*, participants looked up different flights across the two-week period using the same customer service number). We provided participants a paper journal in which to track their task responses and to

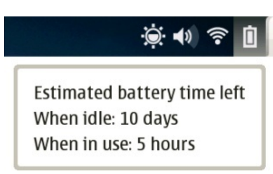

Figure 5: The Nokia N810 battery interface which is displayed when the user clicks on the battery icon found in the status bar.

Table 3: Timeline of user study across the 4-week deployment.

	Day	Study Components
Phase 1	0	**First Meeting**: Study explanation, initial interview, first battery interface deployed
	1	Task on web browsing, survey #1-1
	4	Task on Skype, survey #1-2
	7	Task on listening to music, survey #1-3
	10	Task on watching videos, survey #1-4
	13	Task on both web browsing and listening to music, survey #1-5
	14	**Second Meeting:** Interview on first interface, logs copied, switch to second interface
Phase 2	15	Task on web browsing, survey #2-1
	18	Task on Skype, survey #2-2
	21	Task on listening to music, survey #2-3
	24	Task on watching videos, survey #2-4
	27	Task on both web browsing and listening to music, survey #2-5
	28	**Final Meeting:** Interview on second interface, logs copied, devices and journals collected

answer survey questions. To ensure that the tasks were distributed across the two weeks, we sent the participants the task list via email on a specific day within the study timeline and asked participants to complete the task within 24 hours of receiving it. Table 3 shows a complete study timeline.

Participants answered a set of seven questions before and after each day's task to test their understanding of how the battery of the Nokia N810 drained while they performed the task (14 questions total). The first two questions asked the current time and how much battery time was remaining. For questions 3-7, participants rated their predictions on whether they could complete a specific task (*e.g.*, "You want to watch a 2.5 hour movie on your Nokia N810. Is this currently possible on your Nokia N810 while running on battery?"). For these questions, the participants rated responses using the following Likert scale: 1 - Definitely Not; 2 - Probably Not; 3 - Maybe; 4 - Probably; 5 – Definitely. For consistency, the questions were kept the same before and after each task and were the same across all tasks. Also, each task was definitely possible to complete on a fully charged battery and thus the correct answer depended on users' ability to interpret the battery time remaining interface. We note that the hypothetical task scenarios in the pre- and post-task questionnaires were all different from the study tasks we asked them to complete.

We met with participants a total of three times to conduct interviews at the beginning, middle, and end of the study. The interview questions focused on their current knowledge of battery interfaces, their thoughts on battery usage rates, and, after using the device, their specific thoughts on the battery usage rates and their confidence in battery interfaces.

To verify that participants attempted all tasks and to determine the accuracy of their responses, we installed logging software on all deployed N810s. The logging software recorded time-stamped battery readings along with a list of the applications currently running on the device every 10 seconds. The logging software did not track how the participant used the applications, but only that it was active. Although the logging software consumes battery power, because our models were built with the logger, it is consistent in the study. The logs allowed us to confirm the accuracy of the participants' responses about how much total battery time remains on the device and whether different tasks can be performed on the device. Because the traces used to build our battery usage profiles were gathered with the N810's screen brightness and audio volume at 50%, the logging software also enforced those settings to remain consistent with our traces. Additional traces with different hardware settings can be gathered in the future to determine how different settings impact the battery usage.

Participants

For this study, we aimed to recruit a diversity of people, with an equal distribution of males and females and a range of ages from 20s to 50s. Initially, we recruited 10 participants from Toronto, Canada and Seattle, USA (5 from each city). We counter-balanced the study, with half of the participants using TCBI in the first two-week phase and half using the Nokia interface first. Participants were randomly assigned to one of the two groups (TCBI-First or Nokia-First). We recruited participants from our pool of survey participants, via word-of-mouth, and through postings to the online classified website Craigslist. Participants were compensated $60 for completing the entire 4-week study, with $30 given after 2 weeks and $30 at the end. Due to attrition, we ended up with 8 total participants (see Table 4) whose data we could use to serve as a comparison. The two participants who dropped out were both in the TCBI-first condition, which lead to a slightly uneven counterbalancing.

Results

The results of our user study show promising trends for the Task-Centered Battery Interface. In this section, we present the findings from both the quantitative and qualitative aspects of the user study. In particular, we focus on general usage, how TCBI improved both understanding of battery usage and confidence in the information provided, and participant suggestions for improvement.

General Usage

The final eight participants in our study completed all of the tasks and questionnaires on days distributed throughout the duration of the four-week study. Many of them used the Nokia N810 for activities beyond the staged tasks, including browsing, email, Internet radio, and playing built-in games. One participant (DP5) also customized her device with a new skin and wallpaper and explored new applications for download. There were some participants who only used the device for completing the tasks, and not surprisingly, we saw more use of the device in the first portion of the study while it was still new to the participants. When asked about charging patterns, most charged the device as-needed, while a few charged daily.

Table 4: Participants and demographics for the 8 final participants from the TCBI deployment study.

ID	Gender	Age	Occupation	Computer Experience	City	First Interface
DP1	M	20-29	Student	Exp.	Toronto	TCBI
DP2	M	20-29	Student	Exp.	Toronto	Nokia
DP3	F	20-29	Student	Inter.	Toronto	TCBI
DP4	M	30-39	Child care	Inter.	Toronto	Nokia
DP5	F	30-39	Health clinic worker	Exp.	Seattle	TCBI
DP6	F	50-59	Unemployed	Basic	Seattle	Nokia
DP7	M	20-29	Student	Exp.	Seattle	Nokia
DP8	F	30-39	Homemaker	Inter.	Seattle	Nokia

Understanding of Interfaces

We measured the correctness of responses from the participants' journals by corroborating them with the data from the logging software installed on the device. The start time from the pre- and post-task questionnaire provided the boundary markers to identify when the participants were performing their tasks. Extracting the voltage levels at the respective timestamps allowed us to infer the amount of time remaining for different tasks (*e.g.,* video, web browsing) based on our model. We used these numbers to determine if the responses to questions 3-7 were correct.

For affirmative answers to the questions (*e.g.,* Yes, I could do a 1 hour Skype conference call), we considered the responses to be correct if the participant answered Probably (4) or Definitely (5). Negative answers (*e.g.,* No, I could not do a 1 hour Skype conference call) were considered correct if the participant responded Definitely Not (1) or Probably Not (2). We scored Maybe (3) responses as correct if the correct answer was within a 30 minute time window, but there were only a handful of these responses. Using this method, we found promising results, which indicate that users seem to provide more correct responses using TCBI than they did with the standard Nokia interface. This suggests that participants had a better understanding of their device's battery life and could make better task use estimations. Overall, participants answered 94.8% of the questions correctly using TCBI and 73.5% using the Nokia interface (see Figure 6). A 2-tailed, paired T-Test indicated that this difference is statistically significant ($p < 0.001$).

In general, interviews with participants confirmed findings that TCBI was more helpful than the Nokia interface. Most said it was more helpful in answering the pre- and post-task questionnaires, and they would like to see this type of interface on other types of devices.

> *"It totally works. I can leave my house without charging the thing, and I don't have to worry about it dying."* ***–DP6***

> *"And for the first one [TCBI], it actually felt like when I was checking it… I felt like I was getting more information. At least I was getting a better idea."* ***–DP3***

Others felt that TCBI helped them understand how different activities drain the battery. Even though we did not have them do any GPS tasks, several noticed that it would drain the battery the fastest, indicating that TCBI helped them develop a mental model of application battery use without having to experience it firsthand.

> *"And I had no idea how consuming they are. The GPS one surprised me how much power it seems to take." –**DP4***

Confidence in Interfaces

We determined the participants' confidence in TCBI through analysis of the pre- and post-task surveys and by asking questions specifically regarding their perceptions of the two interfaces using the confidence responses in questions 3-7 in the pre- and post-task surveys. On these surveys, participants rated their confidence on whether they could do certain tasks with the interface on a scale from 1 to 5 (1 - Definitely Not; 2 - Probably Not; 3 - Maybe; 4 - Probably; 5 – Definitely). We rescored responses based on the confidence of the answers, with 5 and 1 being given a value of 2 points, 4 and 2 being given a value of 1 point, and 3 given a value of 0 points. We then averaged the overall value for each interface.

The results suggest that all participants were more confident in their responses while using TCBI, with an average confidence value of 1.76 (with 2.0 being the maximum), compared to an average of 1.31 using the standard Nokia battery interface. A two-tailed T-test indicated this difference in averages to be statistically significant ($p < 0.001$). Figure 7 shows the individual averages for all participants in the study. Interview responses confirmed that users felt more confident and trusting of TCBI's interface. In general, participants reported that the Nokia's interface did not seem very accurate. For example, several participants noticed after a 30 minute task, the time remaining would actually increase rather than decrease.

> *"I definitely noticed that it [Nokia interface] didn't seem very accurate. For example, there was one time when I started it and I looked and it had 2 hours left and then after I had completed the task (which was about half an hour later), it said I had 6 hours left, so I had used battery but somehow had tripled the amount of battery time I had left." – **DP3***

> *"It [Nokia interface] would give contradictory information. My initial measurement, it would give me 4 hours, but then after I had completed the tasks, it gave me 6 hours." – **DP7***

Several other participants explicitly stated that they were more confident in their responses while using TCBI.

> *"It [TCBI] was definitely more informative than the first one, and I felt like I could answer the questions that were being posed with more confidence." –**DP1***

> *"It seems that I got a lot less certainty out of the second one [Nokia interface]. So after going from the first one [TCBI] to the second one I was very leery, like "what does 'one hour' mean? but I think that might just be a product of having my information taken away from me" – **DP2***

> *"It [TCBI] was significantly easier to confidentially say yes or no to the questions." –**DP4***

Finally, two curious participants decided to test the accuracy of TCBI to see if they could trust it by using an external timer to compare the drainage on the interface. Both determined that the times reported by TCBI were indeed accurate.

> *"I felt like I could mostly trust it. I tested it out once, just to see if it was really accurate down to the minute. It was, it said in 2 hours the battery would go dead, and in like, 2 hours and 1 minute, the battery went dead." –**DP5***

> *"I even left it on for a while [to drain the battery] just to see if it worked, to see if the new interface showed smaller and smaller numbers. And it did!" –**DP6***

Suggestions for Improvement

In the mid-study and post-study interviews, we asked participants to provide feedback on both interfaces and how they could be improved. Although all participants reported TCBI to be an improvement on the standard Nokia interface, there were many suggestions for how we can improve the interface design even further. Nearly every participant asked to increase the font size on TCBI, as it was somewhat difficult to read on the small resolution screen of the N810. Many felt that it would be acceptable for the interface to take up nearly the entire screen in exchange for a larger font size and larger icons. Another participant noted that the battery meter for each application were not overly useful, and suggested that only one be shown.

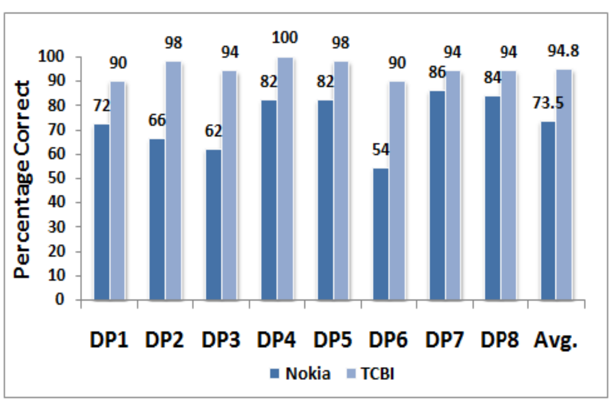

Figure 6: Average percentage of correct responses to pre- and post-task questionnaire.

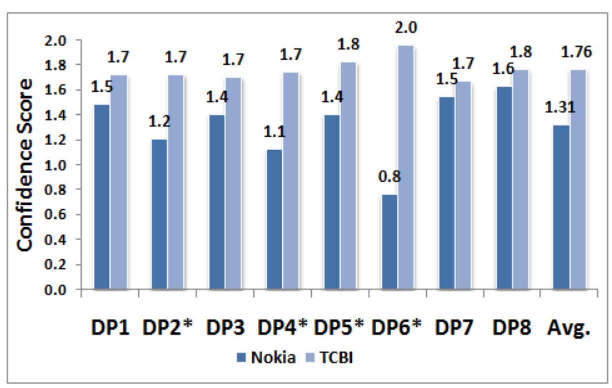

Figure 7: Average confidence scores for all participants in the user study. Statistically significant comparisons ($p < 0.05$) are marked with an asterisk.

*"The icons and the words are a bit small. And the status bar, the green and the red next to the individual application, I don't know if that really communicates well, The only place I would keep that bar is in the current applications at the top." **–DP8***

There was some concern that the TCBI screen was too cluttered. Several suggested that more interactivity may be helpful to explore different application combinations.

*"Because it wasn't interactive, it wasn't easy for me to say stuff like 'I want to look at music and GPS at the same time'… it would have been nice if it was interactive so I could turn things on and off and test to see what would happen instead of actually having to do the programs" **– DP2***

Finally, several participants wanted a more glanceable display. The TCBI icon only had a green dot that did not change, unlike the Nokia interface which showed a battery draining over time. Several suggested making the icon look more like a battery changing over time. Another participant suggested TCBI as a widget integrated into the desktop.

*"It didn't have a display on the [task bar]. It would be nice if the little green dot displayed some sort of information like a traditional battery would." **–DP2***

*"Maybe if you could make it a widget, that would make it so much more convenient. I kept having to close it before I could open anything else." **– DP6***

DISCUSSION

Although this work was explored on the Nokia N810, a wireless Internet tablet, the approach we used can be re-applied on other mobile computing platforms, such as cell phones or laptops. In this section, we discuss how to improve the usability, robustness, and generalizability of this work.

Enhancing the Usability of TCBI

Deployment results indicate that TCBI could be improved by making the information more interactive, glanceable, and less cluttered. Although participants liked being able to learn how potential application usage context impacts the battery time remaining, the current interface does not allow the user to specify envisioned task scenarios. Instead, TCBI currently uses a large amount of screen space to render the current application context and how the usage of additional applications impacts the remaining battery time. As a result, the user must incrementally run a subset of the applications before the interface shows the desired information. Future versions of the interface should allow the user to plan ahead and query the interface for battery time remaining for a specific set of applications. For example, clicking on the set of applications she intends to use, or indicating that she needs Skype for 10 minutes and wants to know how much time would remain for watching videos.

By removing the portion of the screen displaying how the usage of one additional application would impact remaining battery time, the interface itself will become less cluttered. We then would be able to display different visual elements at a bigger size, thereby enabling users to quickly glance at the interface. Presumably, the user already has an awareness of the applications running on her device. Thus, requiring the user to open TCBI to understand how much battery time remains for that particular device usage context is unnecessary. In future implementations, we intend to make the battery icon reflect the estimated battery time remaining for the current set of applications running without forcing the user to launch TCBI itself.

Finally, we believe there should be a way to indicate when the model is more prone to error. These errors could have a big impact on the device usage, especially when the amount of battery charge remaining is low. As shown in Figure 4, the error rate of the model varies depending on the voltage, and thus we could include simple indications of the error ranges. For example, when the estimation is more error-prone, TCBI could show it in a different color or style.

Enhancing the Robustness of TCBI

TCBI estimates the amount of battery time remaining by first computing a battery usage profile for each application. These battery usage profiles are produced from battery traces we manually gathered by using one application at a time on an N810 with the N810's screen brightness and audio volume set at 50% and a fully charged battery until that battery dies. This approach would be impractical to estimate battery time remaining for all existing and future applications. In addition, in real life, hardware settings on a user's mobile device will be adjusted to the user's preference. Finally, this approach does not take into account the diminishing of the battery cycle over time. One way to address these problems is to use crowdsourcing to gather a large number of battery traces over a wide range of hardware settings and stages in the battery's lifetime. We envision that as people use their devices, when a common application is used by itself, it can log time-stamped battery readings, hardware settings, and the battery's cycle count. The device will upload these partial traces to a database. With a large enough user base, it is possible to group partial traces with similar hardware settings and battery cycle count to produce a battery usage profile. Accounting for different hardware settings and battery cycle count can potentially improve the accuracy of battery time estimates.

Generalizing TCBI to Other Platforms

Other similar commercial products such as EZ Battery Life [8], Battery Time [6], BatteryGo [3], myBattery [13], Battery Juice [4], Battery Magic [5] and so on, also attempt to inform the user how long she can use one application on a mobile device. Those products are limiting in two ways. First, our lab studies showed their predictions to be fairly inaccurate by up to 1 hour. In comparison, TCBI estimates how long a person can use a single application to ~738.4 seconds (12.3 minutes) on average. Moreover, the commercial products only assume that one task will be performed at a time. On many mobile computing platforms, users expect to be able to perform multiple tasks at once, and several participants mentioned wanting a TCBI type of interface on other devices. Our current TCBI implementation not only works for single application usage scenarios, but also provides estimates for multi-application usage cases. This is enabled by the development of a multi-application battery time remaining estimation function which is accurate to ~769.3 seconds (12.8 minutes) on average. Although we propose here one function for estimating battery time remaining, we believe it is worthwhile to investigate further ways of estimating how application usage impacts battery time remaining and how to make this approach work on more complex devices (*e.g.*, laptops). However, the difference between laptops and Internet tablets and phones is that laptops have many more processes. Specifically, our approach does not look at background processes running on a device. This might be addressed by allowing per-device training to make models more accurate to account for these background processes.

Study Limitations

Although our study has promising results, there are a few limitations we believe should be discussed. First, we only had 8 participants in our study from two North American cities. We had a diversity of participants, but it is possible the small number of users were not representative of all mobile device users. We were still able to produce statistically significant results, backed up heavily by qualitative data. Second, because participants dropped out of the study, we did not have a perfectly counterbalanced study design, with 5 using the Nokia interface first and 3 using TCBI first. Third, the tasks performed and the pre- and post-task questions may be more structured than normal routines. However, we designed tasks to match common real-world scenarios. Finally, we calculated the participant accuracy results based on our own model of battery life remaining, and thus there could be bias toward TCBI. Our model, however, has been validated to be within 12~17% error on average, and we were conservative in assessing correctness.

CONCLUSION

We described the design, implementation, and evaluation of a task-centered battery interface for mobile devices. The goal of TCBI was to help users better understand how different applications affect battery usage. Our implementation of TCBI for the Nokia N810 shows users more details on the remaining battery life left while popular applications are in use (*e.g.,* Web browser). TCBI is agnostic to the mobile platform and could be used across a variety of devices (including the recently released N900 which includes a phone feature). We provide estimations for multiple applications. To test how well TCBI improved user understanding and confidence, we conducted a user study comparing it to the standard Nokia N810 battery interface. Analysis shows users preferred TCBI, had more confidence in estimations, and had a better understanding of time remaining for other tasks. Results from this research provide researchers and designers with initial insights for how to build more accurate and useful battery interfaces.

REFERENCES

1. Anand, M., Nightingale, E. B., and Flinn, J. 2003. Self-tuning wireless network power management. In Proc. of MobiCom '03, 176-189.
2. Banerjee, N., Rahmati, A., Corner, M. D., Rollins, S. and Zhong, L. Users and Batteries: Interactions and Adaptive Energy Management in mobile systems. In Proc. of Ubicomp 2007.
3. Battery Go! http://appshopper.com/utilities/battery-go (accessed Mar 11, 2010)
4. Battery Juice. http://appshopper.com/utilities/battery-juice (accessed Mar 11, 2010)
5. Battery Magic. http://appshopper.com/utilities/battery-magic-pro (accessed Mar 11, 2010)
6. Battery Time. http://widgets.yahoo.com/widgets/battery-time (accessed Mar 11, 2010)
7. Bloom, L., Eardley, R., Geelhoed, E., Manahan, M. and Ranganathan, P. Investigating the Relationship Between Battery Life and User Acceptance of Dynamic, Energy-Aware Interfaces on Handhelds. In Proc. of MobileHCI 2004, 13-24.
8. EZ Battery Life. http://appshopper.com/utilities/ez-battery-life (accessed Mar 11, 2010)
9. Flinn, J. and Satyanarayanan, M. 2004. Managing battery lifetime with energy-aware adaptation. ACM Trans. Comput. Syst.22, 2 (May. 2004), 137-179.
10. Harter, T., Vroegindeweij, S., Geelhoed, E., Manahan, M., and Ranganathan, P. 2004. Energy-aware user interfaces: an evaluation of user acceptance. In Proc. CHI '04. 199-206.
11. Iyer, S., Luo, L., Mayo, R., and Ranganathan, P. 2003. Energy-Adaptive Display System Designs for Future Mobile Environments. In Proc. of MobiSys '03, 245-25.
12. Miettinen, A.P. and Hirvisalo, V. Energy-efficient parallel software for mobile hand-held devices. In Proc. of HotPar 2009.
13. myBattery.http://www.appleiphoneschool.com/category/mybattery/ (accessed Mar 11, 2010)
14. Pering, T., Agarwal, Y., Gupta, R., and Want, R. 2006. CoolSpots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In Proc. of MobiSys '06. 220-23.
15. Rahmati, A., Qian, A., and Zhong, L. 2007. Understanding human-battery interaction on mobile phones. In Proc. of MobileHCI '07, vol. 309, 265-272.
16. Ravi, N., Scott, J., Han, L., and Iftode, L. Context-aware Battery Management for Mobile Phones. PERCOM 2008, 224-233.
17. Sorber, J., Banerjee, N., Corner, M. D., and Rollins, S. 2005. Turducken: hierarchical power management for mobile devices. In Proc. of MobiSys '05. 261-274.
18. Vallerio, K. S., Zhong, L., and Jha, N. K. 2006. Energy-Efficient Graphical User Interface Design. IEEE Trans. on Mobile Computing 5, 7 (Jul. 2006), 846-859.
19. Zeng, H., Ellis, C. S., Lebeck, A. R., and Vahdat, A. 2002. ECOSystem: managing energy as a first class operating system resource. *SIGPLAN* 37, 10, 123-132.