Interviewing Peripheral Display Designers and Developers

Tara Lynn Matthews

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2005-19
http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-19.html

November 28, 2005

Acknowledgement

I would like to thank my advisors, Jennifer Mankoff and Anind Dey. This work was supported by NSF grants IIS-0209213 and IIS-0205644.

Interviewing Peripheral Display Designers and Developers

Tara Matthews
EECS Department
UC Berkeley
Berkeley, CA 94720-1770, USA
tmatthew@cs.berkeley.edu

ABSTRACT

Traditionally, computer interfaces have been confined to conventional displays and focused activities. However, as displays become embedded throughout our environment and daily lives, increasing numbers of them must operate on the periphery of our attention. *Peripheral displays*, ubiquitous computing devices that present information without demanding attention, are difficult to build, particularly because they must dynamically manage the cognitive load they place on users. We have worked to plan and create a toolkit, the Peripheral Display Toolkit (PTK) [5] that focuses on supporting the implementation of peripheral displays. In order to continue to inform future improvements to the toolkit, we conducted interviews of peripheral display designers and developers. Interview results point towards a need for the support of early design decisions, implementation support for gathering and analyzing input, and evaluation. This paper summarizes the interview methodology and results, and discusses the results in search of future research directions.

1. INTRODUCTION

Traditionally, computer interfaces have been confined to conventional displays and focused activities. However, as displays become embedded throughout our environment and daily lives, increasing numbers of them must operate on the periphery of our attention. *Peripheral displays* are ubiquitous computing devices that give information to a user without demanding their full attention. This allows a person to be aware of more information without being overburdened by it [1]. Peripheral displays are difficult to build for several reasons: they must dynamically manage the cognitive load they place on users; they may display information ranging from low to critical importance; they can get input from almost any source; they can display output in virtually any modality; and they are often distributed across multiple machines.

To alleviate the burden of implementing peripheral displays, we created the Peripheral Display Toolkit (PTK). In addition to support for remotely distributed applications and the easy connection of inputs to outputs, The PTK architecture adds support for three key characteristics of peripheral displays: abstraction, notification, and transitions. *Abstraction* is used to transform incoming data to meet the needs of an output device. A designer can indicate the *notification* level of incoming data, which corresponds to its importance. *Transitions* are used to update the output display to attract an appropriate amount of attention based on the notification level.

To inform future research related to the PTK or other support for peripheral display creation, I have conducted interviews of peripheral display designers and implementers. The goal of these interviews is to determine the best way to support the creation of peripheral displays, whether through design, implementation, or evaluation. The results of the interviews are discussed in this paper. Briefly, the results point towards a great need for the support of early design decisions, implementation support for gathering and analyzing input, and evaluation. The results confirm that design decisions related to the key features supported in the PTK (abstraction, notification, and transitions) are one of the most challenging parts of peripheral display creation, but it is clear that the PTK's code support does not alleviate the design challenge of choosing how to employ them. Results also show that evaluation, especially of early prototypes and methods for measuring particular metrics, was a clearly deficient area in current research.

1.1. Paper Overview

The remainder of the paper discusses the interviews. First, I describe the methodology of selecting interview participants, creating questions, and conducting interviews. I also describe the participants' backgrounds. Then I present the results of the interviews, summarizing responses to interesting questions, discussing the areas of disagreement among participants, and selecting other interesting comments from participants for discussion. Following this, I discuss the future direction of the PTK or other related research given the interview results. At the end of the paper, I present an appendix with links to literature survey documentation and interview questions.

2. INTERVIEW METHODOLOGY

Over a one-month period, I conducted interviews with ten peripheral display designers and implementers. This section discusses the process, from determining whom to interview to transcribing interview audio recordings.

2.1. Literature Survey and Participant Selection

The first step in conducting interviews was selecting participants. To this end, I read all the papers I could find on peripheral displays and looked at papers related to peripheral awareness (*e.g.,* notification and interruptibility research). I developed a list of all the key people working on these projects; there were 49 people on the list. Since I did not intend to contact so many people, I developed criteria for participants. Namely, participants should have designed and/or built a peripheral display (including ambient or notification displays). People who had not designed and/or built a peripheral display would not have the experience necessary to give feedback on what is needed to support the creation process. For example, they would not know what is the hard part of designing or implementing a peripheral display, what difficulties they had with code or hardware, how users reacted to their displays, *etc*. Also, I believed that people who had designed anything with a consideration for peripherality, not necessarily using computers, would be excellent resources. Therefore, I was sure to include some people who had less technical backgrounds. I believed that such people would be more focused on the design considerations that made their displays peripheral. I used these criteria to rank my list of 49 people.

With my ranked list, I contacted only those who had designed and/or implemented a peripheral display – 22 people total. Of these 22, 12 people responded and I interviewed 10.

2.2. Participants

The interview participants were evenly spread between industry research (4 people) and academia (6 people). No more than two people came from the same institution, and eight institutions were represented. Three of the participants were primarily designers, three were primarily implementers, and four were both.

Participants had a self-reported average of 5.3 years experience designing and/or implementing peripheral displays. They had been involved in the creation of between 1 and 20 displays, with a median of 3 per person. Three participants had built 1 or more toolkits relevant to peripheral displays.

Participants self-reported their design and implementation expertise on a scale of 1 to 5, where 1 was "no experience" and 5 was "very experienced (3+ years, multiple displays built)." Their average ratings were 4 for design and 3.3 for implementation.

2.3. Interview Format: Phone and In-person

Interviews were conducted in-person when possible and over the phone otherwise. The interviews were 50 – 80 minutes long. After collecting consent and audio release forms, I recorded the interviews with a digital audio recorder[1].

Each interview began with me explaining my high-level goal: I wanted to determine the ways in which to support the process of designing, implementing, and/or evaluating peripheral displays. I then asked my carefully designed questions (see Appendix B). Often, participants would answer questions from my list without me asking, or say something that would indicate that a question would not be relevant.

In designing the interview questions, I tried to make them open-ended, not lead, not impose my way of thinking about peripheral display creation, and not show too much of my opinion. The topics covered by the questions were background, design, implementation, input, interruptibility, user customization, evaluation, difficulties, and toolkit requests. I tested the questions by conducting two practice interviews with fellow researchers in my group who had created peripheral displays. I revised the questions with the feedback from each practice.

After finishing an interview, I transcribed the audio recording. I had help from an undergraduate who transcribed one interview, but the rest I transcribed. For these, I only transcribed the sections of the recording that were relevant to the project (which was typically most of the interview). Transcribing an hour-long interview took between 2.5 and 3.5 hours.

In conducting both in-person and phone interviews, I found that there was not very much of a difference in the quality. For one in-person interview, it was nice to see the peripheral display demos and to refer to them as we talked. However, this made the audio transcripts more difficult to understand. For all the in-person interviews, it was very nice to meet the researchers in person; each of them is a colleague in a way, doing the same type of research.

3. RESULTS

This section presents the results of the interviews. The interviews brought out some common themes in peripheral display creation. First, most people agreed that the hardest part of this process was evaluation, largely because it was not well understood how to get the information needed from tests. Several others thought deciding between design options and gathering input data were hard.

[1] One audio recording was lost due to a recorder crash (the author recommends not using an MP3 player / all-in-one device for audio recording interviews). For this interview, I took written notes during and immediately after the interview.

Second, most people agreed that user customization was important for personal displays. Third, people did not agree on the use of notifications or input data abstraction for peripheral displays.

The rest of this section covers these points and others in more detail, including discussions of the parts of design and implementation participants found the hardest, aspects of implementation required specifically because displays were peripheral, the participants' thoughts on evaluation, input analysis, and user customization, issues upon which participants disagreed, thoughts on interruptibility, requests for toolkit support, and interesting points made by participants.

3.1. Hardest Parts of Creating Peripheral Displays

When asked what the hardest part of designing peripheral displays was, the responses were as follows:

- 7 said evaluation
- 2 said selecting from too many design options
- Others: aesthetics, user customization, justifying its existence, details (*e.g.,* smooth interaction)

Clearly, evaluation was noted as the most difficult part of peripheral display design. Obviously, early stage evaluation is highly related to the second hardest part, selecting from too many design options. An evaluation technique that helped wade through design options to select a good one would be extremely valuable. One participant aptly stated the problem, "How do you generate as real of an experience as would suffice for your data collection needs? I'm looking for [an evaluation technique] that will have a high data pay-off for a low time cost." However, such an evaluation technique does not currently exist, as other participants pointed out: "the real value in many of these systems is only apparent longitudinally."

When asked what the hardest parts of implementation were, the responses were as follows:

- 3 said getting input data
- 2 said coordinating outputs
- Others: robust distributed systems, real-time performance, building and using embedded hardware

Though three people complained of the difficulty of gathering input data, two of them explicitly warned that such work is another area of research entirely; people get their Ph.D.s working to solve input gathering and analysis problems of various kinds.

3.2. Implementation for Peripherality

The PTK focuses on supporting peripheral display implementation through a middle-layer of code for features specific to peripheral displays: abstraction, notification, and transitions. However, one of the problems with the PTK is that this code is fairly lightweight and does not significantly decrease the coding effort for peripheral displays. To determine if participants had spent coding effort making their displays peripheral, I asked the following question:

> What parts of the implementation are required specifically because your displays were peripheral? In other words, other than getting some input data and simply displaying it on the output, is there any significant or difficult code between these two steps?

Responses were as follows:

- 2 wrote code to map input to output
- 2 wrote code to coordinate the updates on outputs
- 2 did not write any code specifically to make the display peripheral

The two who wrote code to map input to output stated that this was a minimal amount of code. The two who wrote code to coordinate the updates on outputs did so for differing reasons: one did so to avoid distracting or confusing the user by too many simultaneous updates; the other did so for performance reasons, avoiding CPU spikes and network saturation caused by many simultaneous updates. One of the participants who did not write specifically peripheral code noted that it is primarily how a person uses a display that makes it peripheral. While there are some special aspects in the design, there is nothing special in the implementation.

3.3. Evaluation

It is clear from the interviews that participants found evaluating their peripheral displays a challenge. Though a number of them did fairly extensive evaluations of their displays, many had trouble getting the desired information from these evaluations. The comments made point to some interesting directions for evaluation research for peripheral displays, as summarized in this section.

Following is a summary of the evaluations (or lack thereof) conducted by participants:

- 5 deployed to typical users for long time. These evaluations included the following:
    - Deployed to typical users, in their workplace, for 6 to 8 weeks. Collected qualitative feedback.
    - Deployed in a museum for several weeks.
    - Deployed to thousands of people throughout the company and collected email survey feedback.
    - Deployed to typical users in their homes for several weeks. Collected qualitative feedback.
    - Paper prototyping, informal heuristic evaluation, and both lab testing and field studies. For the field study, deployed to 250 of people throughout the company for two weeks and collected survey feedback.
- 3 deployed to people in their research department.
- 4 did early prototype tests / needs analysis. These evaluations included the following:
    - Wizard of Oz testing by simulating the primary task and then simulating the experience the user would have interacting with the peripheral display. The goal of this study was to get as much information with as little implementation time as is possible.
    - Wizard of Oz testing on 4 to 5 peripheral display prototypes.
    - Ethnographic interviews with users and participatory design.
    - Diary study of information workers used to inform a peripheral display used for multi-tasking and task switching.
- 1 did no evaluation.

Despite the fairly high number of displays that were deployed for long periods of time, a number of the participants had trouble getting the information they wanted from their evaluation. The main problem is that the metrics people want to measure are inherently difficult to measure. For example, several people found that surveys were not satisfying: while they could ask "was the display distracting," they did not fully trust the self-reported answer.

The common questions about evaluating peripheral displays were as follows:

1. How does one get quality user feedback before completely implementing displays?
2. How does one measure desired metrics during user deployments and tests?

More specifically, users made the following requests and suggestions for evaluations they would like or that they thought would be effective for peripheral displays:

- Wizard of Oz is needed to get as much information with as little implementation time as is possible (this participant had conducted such evaluations).
- Evaluation is needed at the proof of concept stage, though it is unclear how.
- Want to be able to compare two or more types of displays based on appropriate metrics (*e.g.,* displays on a table versus on a wall).
- Non-quantitative, non-numerical approaches to evaluation, a la Bill Gaver.
- Devices should have logging mechanisms (*i.e.,* a way to record and save their own histories). "This helps you understand these devices as they are embedded into people's lives [as is the Technological Probes paper from CHI '03 [2]]."

Participants named the following metrics as important to measure during evaluations:

- 4 said appeal, desirability, aesthetics
- 3 said usefulness, information conveyance (*e.g.,* is the display saving users time during the day, is it making them aware of the information it displays, *etc*.).
- 3 said distraction from primary task
- 1 said a combination of the interruption, reaction, comprehension (IRC) [4]

On participant did not have anything to add to the list, saying,

> "I would have liked [to use some metrics], if I had known what metric to look for. That I guess is where I felt there was a lag in the project… At least we did not know of enough, or of any psychological theory that could come and assist us here: something that you could measure and then predict about longitudinal effects. Do you know of any?"

A common theme throughout most interviews was that evaluation is the hardest part of creating peripheral displays. Part of the reason it is difficult, is that peripheral displays commonly involve hardware so building a realistic enough prototype is time consuming. One participant noted how the iterative design cycle is lengthened,

> "How do you evaluate a peripheral display? You can't do a typical lab, usability study. We had to implement a working prototype and deploy it in people's work place. If we had found that it was all wrong, we would have had to throw away all that work."

Another reason is that peripheral displays are fundamentally different from primary task applications, for which most evaluation techniques have been created. Realizing this problem, one participant commented,

> "Fundamentally, most technology that is out there is about maximizing efficiency. And I think that a lot of ambient display stuff is not about maximizing efficiency. Then you have all of these paradigms from Jakob Neilson, and what they basically say is how easy is it to do this thing, how efficient is it. When that is not necessarily a fundamental metric, I think you have to reevaluate those systems of evaluation."

Also, peripheral attention is not well understood and difficult to measure, as another participant explained,

> "[Evaluation] is so hard when you are talking about peripheral awareness because in some ways it is like housecleaning in that people cannot say whether or not it is done because they don't notice it. It is only if it is bad that you have something to say. It is the same with a successful ambient display. How are you going to prove that it is successful, except if people after two years are still using it? Then there must be something about it. But you cannot test it after a month because it is like a language: you have to learn and you are not learning it in an intellectual way, you are learning it in a sympathetic way. You are getting it internalized gradually."

Despite these challenges, peripheral display evaluation is possible. As summarized above, the participant suggestions point to several interesting areas for future research. Also, one participant was successful in conducting several studies for multi-tasking situations. These studies were used to evaluate notification awareness and delivery, which is very relevant to peripheral displays. The format was an in-lab study that involved building a dual or triple task scenario. The metrics measured were overall primary task time, time to notice a notification, time to respond to a notification, time to abandon a notification and reacquire the primary task, errors in terms of notifications missed, and memory for notification content (either recognition memory or recall). The participant emphasized the power of measuring recall and recognition memory because "if the display is not elegantly designed, you will see a significant drop in memory performance for the content that was displayed peripherally." To measure recall and recognition memory on peripheral displays, one could use a word-stem completion study like the voyeurism study from CHI 2003 [10]. This study used implicit metrics by testing word fragment completion.

With further analysis and experimentation, further evaluation methods could be developed to measure the metrics relevant to peripheral displays at various stages in the creation process.

3.4. Input Analysis

Peripheral displays often involve displaying information in an easy to glance at and understand way. For this reason, these displays may benefit from input processing or analysis in order to distill or aggregate information. Also, peripheral display applications may wish to use input that requires analysis in order to make high-level inferences. For example, in a remote presence application, video analysis may be needed to determine how many people are in a remote location and when they are moving. To explore the need for input analysis, participants were asked, "Are there any complex input processing or analysis you've wanted to do for a peripheral display that you didn't because it would take too long? For example video, audio, text, or numerical processing or analysis?" A summary of the results is as follows:

- 1 said no, I have done either none or very simple input processing
- 1 said no, I have done this by faking it
- 4 said yes, but I haven't done any
- 3 said yes, and I have done input processing/analysis

The types of input processing and analysis people have done in their displays include

- audio and video analysis to determine activity in a room (determined change levels),
- image analysis (to find faces) and visual feature extraction (to get only the faces), and

- video processing to modify video based on emotional cues derived from the environment (then re-outputting it in another environment)

Other people wanted or planned to implement the following types of input analysis:

- general analysis of Web or XML Web-based service data,
- audio and video analysis to determine activity in a room (*e.g.,* count the number of people, figure out who and where people are), and
- a Windows event logger that does complex time-series pattern discovery and visualization to automatically track workflow and recognize recurring task patterns.

Two people commented that, even though they would like input processing or analysis support, this is another discipline and people get Ph.D. writing related systems. One of these two said that mocking difficult to generate input for evaluations is very effective. These points are important to consider before building complex input analysis support into a future toolkit iteration.

3.5. User Customization

When asked if any of their displays allowed users to customize input data or the way in which data was displayed, participants responded as follows:

- 6 said yes.
- 2 said no, but it is important and we want it or are working on it.
- 2 said no, and it is not important or is not applicable in our display.

Most participants emphasized that user customization was extremely important. One participant stated, “One of the only things that would make [the peripheral display] work was giving the user total control over it.” An interesting lesson learned from one participant was that there is such a thing as too much customization. In one of his displays, the user had to choose all the input content, assign importance levels to it, and determine the layout of the content on the output display. Users did not respond well in evaluations, saying it was not useful.

Of the people who did not think user customization was applicable, one had created public displays but thought it would be important in personal displays. Another participant commented that customization is only necessary in deployed versions, a stage he had not reached.

3.6. Contention in the Field: Notification and Abstraction

Participants disagreed on two issues: the appropriateness of notifications and of input data abstraction. These issues brought to light a problem with our definition of peripheral displays and the scope of the PTK.

Regarding notifications, participants were asked, “Did you consider including alerts on any of your displays? For example, attempting to attract the user’s attention when important data arrives. Why or why not?” This question comes down to the following: how peripheral must peripheral displays be? Should peripheral displays always remain in the periphery of our attention or is it appropriate for them to enter our focal attention? The responses were varied:

- 3 said alerts do not belong
- 1 said only mild visual alerts are OK
- 3 said alerts are OK, but not applicable for my project
- 2 said alerts are important

Coming into the interviews, I firmly believed that notifications are important to peripheral displays. How could you keep something in the periphery if you could trust it to let you know when something important happened? But as several participant argued, perhaps important information that is meant to be in your foreground should not be displayed on peripheral displays.

> "I would argue that the question is not whether a peripheral display should try to grab your attention, but the question is what display should be used to convey a particular piece of information. So I would argue a peripheral display should be for peripheral information. But if there is something that needs to be seen, it should be conveyed differently. It should be on the display that is used to get your attention. Let the peripheral display be peripheral, don't let the person ever get frustrated with it."

This quote brings out the tension caused by the term "peripheral" standing for notification *and* ambient displays *and* other peripheral visualizations. Perhaps grouping these three together makes the definition of peripheral displays too broad. It may be more difficult to address the problem of supporting their creation when these related, but different types of displays are grouped together. Also, it is confusing to describe notification displays as peripheral. After all, the name peripheral display is misleading if it is used to convey focal information, a point made by one of the PTK UIST reviewers. In addition, the combination of peripheral visualizations and ambient displays caused some confusion. This point is further proven by the following conversation:

> (*ME*) Do you believe that notifications and peripheral displays are separate or are they in the same category?
>
> (*INTERVIEWEE*) Your use of "peripheral display" is confusing, because peripheral display could mean a piece of hardware or a piece of software. We don't do much hardware here, so we think of them as the same.
>
> (*ME*) I generally think of peripheral displays as either physical or graphical.
>
> (*INTERVIEWEE*) Yeah, that's my point: that's kind of confusing.

Others thought that peripheral displays inherently require notifications:

> "I thought that was what these displays were doing in general; they were alerting people when data arrives… When data arrives they would bring themselves to the forefront."

There is obviously some incredulity caused by my question here. The designer saw all updates to the peripheral display as notifications. What would be the purpose of letting the user know that data had changed if the user does not perceive it?

While I (and most of the participants) tend to think of notifications as the way in which a peripheral display attempts to attract the user's attention, one participant had a unique view. This person called notifications *interruptions*, and defined them by how much they interrupt the user from her primary task. This definition approaches notification from the user's perspective and is dependent on the primary task (*e.g.,* if the user is surfing the web, her primary task takes less cognitive effort so subtle updates in peripheral information will not interrupt her; therefore there is no notification). My definition takes the importance of the input data and assigns a notification level based on how much attention the display should attract. We are essentially working at the definition from two opposite sides. For the sake of generalizability, I believe that introducing the user's activity into notification setting decisions is too difficult.

The second point of contention was whether or not abstracting input data makes the data more or less difficult to understand. Though no question was asked directly regarding this point, it was clear that opinions varied. Two participants had created displays that mostly presented information literally. The remaining eight participants had created displays presenting information in some form different from its original form; *i.e.,* the information was abstracted such that features were extracted or details were removed. The reason for abstraction is explained in one participant's paper, where it is hypothesized that "abstractions may be preferable to more media-rich representations by providing a better peripheral, non-attention demanding awareness." [9] This appears to be the general consensus among peripheral display designers, who have tended towards very abstract representations of data.

Such abstractions require some learning time, as one participant pointed out: "You cannot test [an ambient display] after a month because it is like a language – you have to learn and you are not learning it in an intellectual way, you are learning it in a sympathetic way. You are getting it internalized gradually." This sounds promising, but as another participant pointed out, abstracted data is not unique: "You can become automatic on anything if you train long and hard enough. It's called automatization." But abstract data can be useful in certain situations: "If the information you want to pop out at the user is a paragraph of text, a simple abstract representation that conveys that exact same content will be better, because it takes too long to read a paragraph of text."

Other logical voices argued the opposite: abstract representations make information more difficult to understand. One participant contended,

> "From a design standpoint, you want to get rid of as many levels of indirection as possible, because each level of indirection requires human cognition to get through it. One reason we want literal is to get rid of as many layers of indirection as possible. So we're doing ambient visual: you have to figure out what the mapping is… and people have to figure that out. The issue with that is: are you going to make the user make it? If no, how do they discover what the mapping is? They have to remember, 'Oh, the number of beach chairs corresponds to the number of people on my buddy list who are present.' [Abstraction] didn't seem to be creating any value other than perhaps aesthetics, but the value of the aesthetics didn't seem to justify all the other costs. So it seemed like aesthetics went up and useful and difficulty of creating and customizing got worse. So that's why we went more literal."

A general criticism of peripheral displays is that they are not useful enough to justify their existence. One of the participants even stated that the hardest part about creating peripheral displays is "justifying their existence." The participant quoted above is touching upon this issue indirectly by claiming that abstraction tends to decrease the usefulness of display. Opponents of this view might ask, does it need to be useful? Can it be calming and aesthetic, and thus be indirectly useful? One participant said, "I think we're going to move away from our information spheres as machines and towards thinking about them as personal statements." In this view, peripheral displays appear to enter into the realm of computer art. Opponents might then argue that peripheral display work is not computer science research, but is design or art.

I tend to wonder myself if peripheral displays should be pushed much further by computer scientists. It is hard to determine if their usefulness is deeper than it currently appears, when the most popular data to display are weather forecasts and stock quotes. If the true benefit of these displays is only realized through an exploration of the calming or inspiring effects an aesthetic information display can have, should this not be explored by designers? One participant offers a bit of hope:

> "A fundamental rule of media is that when you develop a new media, whatever it is, the first thing you do is recreate the old media in the new media. When you look at the first television, it looked like radio, with people standing around on stages reading from scripts… (31:20) I think ambient media is still at that stage: we're still using it for other information, like stock market tickers."

In my opinion, there seem to be times when abstract representations are appropriate and when they are not. It seems that the real power of abstraction is reducing large amounts of data to an easy to glance at representation. In other words, abstraction can be used to convey aggregated data. Perhaps peripheral displays would be more compelling if, instead of abstractly showing weather forecasts and stock quotes, they displayed things that are not easily shown with one or two numbers. Perhaps there would less of a clash between computer scientists and designers if these displays had more of a reason to exist than aesthetics alone. One participant commented along a similar vein:

> "If you have the visual space to display *something*, it should be *that*. Like if you're going to take up this amount of pixels, just show that actual data, because you're just abstracting away data that is important. Now if you're trying to combing 10 different things and you only have this small amount of space, that's when you want to aggregate."

The question of notifications and abstraction may be the key to defining the scope of the PTK, a point that is elaborated in the Discussion section.

3.7. Interruptibility

When asked if they had considered taking a user's interruptibility into account when displaying output, people answered along the same lines as the notification question: if they thought notification were OK, they thought inferring interruptibility would be nice (but most had not implemented it); if did not think notifications were OK, they thought interruptibility sensing was not applicable. Most said their displays were peripheral enough that they did not need interruptibility sensing.

Several commented that interruptibility inference was hard to do correctly: it is an open research question. Supporting it in a toolkit for peripheral displays may be too broad of a scope. Also, toolkit support implies that we know the right way to do it, which we do not.

3.8. Specific Toolkit Requests

At the end of each interview, participants were asked, "Do you have any specific requests or suggestions for functionality that a toolkit could provide?" The summarized answers are as follows:

- Support for getting input information from the Internet or Web-services.
- Support for evaluation: methodology and metrics.
- Support for rapidly prototyping to test ideas: mapping different inputs for different outputs. (A toolkit may not be useful for creating the final product.)

- Provide ways of assessing large numbers of design options quickly and for choosing the best design.
- Help in determining how to properly visualize the information on the output.
- Support for multiple output modalities (physical, graphical, audio).
- An embedded platform in C with Ethernet, wireless, DHCP, and various built-in inputs and outputs (*e.g.,* sensors, LEDs), that is easy to use and inexpensive.
- Support for testing distributed systems.
- A lightweight way to pass data between machines. Should not need to run a web server on a user's laptop.
- Keep the pipes as big as possible because people will always want to do things you don't expect.

3.9. Interesting Points

The process of interviewing the group of ten intelligent, experienced researchers was very educational. Participants said many interesting things, giving me insight into their research processes. In this section, I highlight and discuss the interesting points that came up during interviews that are relevant to the process of peripheral display creation.

3.9.1. Abstraction as privacy

A few participants commented that abstract representations of information can be used as a privacy mechanism. While it is nice to be able to design interfaces such that a person can walk up and use them, peripheral displays often involve some abstraction that must be explained before it can be understood. When users are able to customize the way in which data is displayed, a properly obscure representation can make data un-readable to others. This is an interesting side-effect of abstraction and user customization explicitly being explored in the Infocanvas [7] project.

3.9.2. Increasing our ability to be peripherally aware

One participant had a particularly interesting quote:

> "[In my noisy apartment building] I could listen to the building and I knew what was going on. I could hear the neighbors and I knew when they were home and when they had guests over – so that wouldn't be a good time to go over. We are able to make so much sense of the most scattered and absurd things, and if I tried to formalize what it was that I was hearing and make a computer model that would do the same for me, it would be impossible. So that gives rise to the idea that we need to redefine the relationship between computers and us… Sometimes people build computers to compensate what people are bad at… I am rather trying to see how they can really further extend what we are good at."

This quote is inspiring because it hints at how peripheral displays can be extremely valuable. Peripheral awareness is something that people are good at. We can monitor our environment through simple cues, like people coming and going, the smell of coffee from the kitchen, the sound of a car pulling into the drive, *etc*. What if these cues were more complex? Could technology be used to supply peripheral "landscapes" that convey large amounts of data? In what situations would this be useful?

3.9.3. Giving us abilities we do not have

Another option is to use peripheral displays to augment abilities we do not have or are not good at. A good example of this is the Stanford iRoom Event Heap [1] display. This display shows a colored circle for all events sent through the iRoom Event Heap. The type of event dictates the color of the circle, and its time to live dictates the size of the circle. Evaluations have shown that the display aids iRoom users in debugging applications and machines running in the room, by visualizing something that people could not normally see. Are there other complex pieces or aggregations of information that we cannot see but that could help us perform certain tasks?

3.9.4. Comparison of output modalities

The interviews drew out the advantages and disadvantages of different display modalities. There is a real reason for these displays to be physical (versus graphical), aside from trying to meld into the environment: people will not tolerate losing screen real estate to something that is not vital to their task. One participant commented about her experience with a peripheral display:

> "When I saw [Sideshow [3]], I said 'Wow, I totally want that on my computer!'… Then I downloaded it and… after two days I stopped using it completely. The reason was that I just wasn't willing to give up the screen space. Even though it only took up this much of your screen [points to a small strip of the monitor], that is still a lot of space, especially when… you have a lot of windows open."

In order to be useful at all, users must be able to peripherally monitor their displays. Graphical displays like Sideshow have the problem that they must take up screen space in order to be visible. On the other hand, physical displays are typically only capable of showing one piece of information at a time: they are not as flexible as a graphical display. Audio provides an alternative, but it often is more interruptive. Some people cannot handle continuous audio, and non-continuous audio is by nature a notification. One participant noted that her users were very emotional about the use of continually changing visual- versus audio-scapes: about half loved visual and hated audio, and vice versa for the other half.

One interesting question is whether or not these disadvantages are absolutes. In other words, is there any way to design a physical display that *is* capable of displaying more than a couple of pieces of information?

3.9.5. Supporting output design is difficult

One of the big challenges with designing output for peripheral displays is that there are several modalities from which to choose and the issues raised by using different output modality will be very different. For example, smell output is a rare output modality that lends itself to peripheral displays, since it tends to be a peripheral sense. Smell output has unique issues:

> "I think [smell output] is a bit complicated because one of the things that makes no difference in other domains, is once the output is out there it stays around for a while, even though you're not doing anything. So if you are using orange to represent something and you squirt some orange out there, the place is still going to smell like orange for a while."

With diverse issues raised by diverse output modalities, how can a toolkit provide general support?

3.9.6. Broken input

One problem that I do not see a solution to is peripheral displays breaking because input sources change formats. For example, if a display gets input by parsing a web page and the page changes it's format a year later, the display will no longer work. To fix the problem would require code changes, something a user could not be expected to do. One participant who had gone through a long-term deployment noted this problem:

> "The only technical reason [the display] has stopped working is that each ticket gets information from a source and the source might change: weather.com changes their format, Outlook changes every 2 or 3 years…"

3.9.7. Awareness vs. disruption

Several people commented about a difficult a design question: how much change should occur in the output to obtain the correct balance between awareness and disruption? Sideshow changed relatively small numbers of pixels at a time. For Infocanvas, the creators found that simply adding and removing items from the display gave the right balance. In Audio Aura [8], the sound-scape had to continually play, or the sounds would be more like notifications than peripheral awareness. For AROMA, the display continually moved even for low or no activity, because otherwise users ignored the display until something happened at which time they were interrupted. The author commented:

> "So what we learned [in AROMA], with the peripheral awareness, there is a very fine balance between when you ignore it completely and when it interrupts you… It seemed to be a major engineering problem that we did not solve, and also a testing problem to figure out how to define this narrow gap where we actually were able to provide peripheral awareness data."

Future work should focus on helping designers find the correct balance of awareness and disruption given the use scenario.

3.9.8. Related research

Two of the participants talked about their research that is similar to the PTK project. First, Microsoft has created a toolkit for sidebar-like peripheral displays to be released with their new Longhorn sidebar [6], which was inspired by the Sideshow project. Any toolkit we build would need to be distinct from Longhorn.

Second, Scott McCrickard and his students are actively working on a project to support the design of notification displays. With the Link-up project [4], they are looking at ways to create good designs before the prototyping stage. Their method involves analyzing user requirements and matching them with elements from systems that have been built in the past. In analyzing user requirements, they use three main metrics that describe the user's situation: interruption, reaction, and comprehension (IRC). The end product will be a library of knowledge that designers can access to get ideas regarding the design of their displays.

4. DISCUSSION AND FUTURE WORK

This section discusses the results presented above in light of future work on the PTK project. First, it explores the definition of peripheral display and proposes a new, narrower one. Then it investigates the areas of possible toolkit support and makes a proposal for the PTK: that it focus on early-stage prototyping and evaluation of peripheral displays.

4.1. A New Definition for "Peripheral Display" is Needed

The first obvious step to make is to revisit our definition of peripheral display and the scope of the project. Currently, we define peripheral displays as "displays that are not at the focus of a person's attention," and we include notification, ambient, and change blind displays in this definition.

However, as discussed previously, participants found the term "peripheral" misleading for notification and interactive displays since these are used to convey focal information. They also had trouble grouping peripheral visualizations and physical ambient displays together. For the PTK, it will be difficult to address the problem of supporting peripheral display creation when related, but different types of displays are grouped together.

There are several categorizations that fall under our definition that are suspect: notification vs. ambient, literal vs. abstract, physical vs. graphical, visual vs. audio vs. haptic vs. smell, visualization vs. display (though I am not sure exactly what the difference is between these two, I think there is something here). While "peripheral display" as we define it includes all of these terms, I believe this is too broad. Under our definition, an Internet browser that periodically refreshes itself would be a peripheral display.

4.2. What Should the PTK Support?

Next, there is a question of what stage in the creation process we should support: design, early prototypes, deployable prototypes, evaluation, or final products. Currently, the stage the PTK focuses on is the early prototyping stage and the deployable prototype stage.

There is a drastic difference between what is needed for a prototype and what is needed for a final, deployable display. Early prototypes bring up issues of design and are in frequent need of evaluation. How should the display show data updates that are very important? Or of low importance? How do we know the users will get the information without undue distraction? Will the users like the way it looks? Is data abstracted in a way that is easy to learn to read? For very early prototypes, it is not necessary for them to be deployed in a real-world situation; heuristic evaluations and lab studies may be sufficient. For later prototypes, deployment in a user's environment is probably necessary. For a final deployable display, physical systems will likely need to use embedded hardware and wireless communication, a graphical system should not be bulky or slow down the user's computer, users should not be required to install Java on their computers, and the display should be robust and attractive.

The key problem with supporting the implementation of peripheral displays (prototypes or final products) as the interviews have shown, is that minimal code is written specifically because the display is peripheral. The majority of code is written to get and analyze input and to display output.

For future work, much needs to be done to support the early prototyping and evaluation stages. Early prototyping focuses on design and is in need of frequent evaluation. This process could benefit from toolkit and methodological support in several ways. A toolkit could make it easier to quickly create a rough prototype to present to users in a controlled environment. The toolkit could then provide mechanisms for dynamically changing aspects of the prototype (*e.g.*, increasing or decreasing the interruptiveness of updates). It could log the state of the display and potentially also interactions the user had with it or her primary task application. I could allow for easy switching between output modalities or input sources. Instead of user customization, which is important for deployments, the toolkit would enable easy designer manipulation for evaluation purposes.

5. CONCLUSION

This paper has discussed interviews conducted to determine the future of the PTK project. The goal was to determine the best way to improve support for the creation of peripheral displays, either through design, implementation, or evaluation. To accomplish this goal, I interviewed ten peripheral display designers and implementers, analyzed the results, and discussed their relevance to the PTK. Results indicated that the hardest parts of creating peripheral displays were evaluation, deciding between many initial design options, and writing code to get and analyze input data. They also showed some disagreement among participants on whether or not notifications were appropriate on peripheral displays, and whether or not it was best to display input data abstractly or literally. In addition, it became evident that our definition of peripheral display was not clear.

These results indicate that future toolkit support should focus on the early prototyping and evaluation stages. Also, the results indicate a need for a clear definition of peripheral displays. Further analysis and thought will be needed to confirm that the proposals for the future of the PTK are appropriate and what steps are needed to accomplish them.

6. REFERENCES

1. Ballagas, R., Ringel, M., Stone, M., Borchers, J. "iStuff: A physical user interface toolkit for ubiquitous computing environments". In *Proceedings of CHI '03*, pp. 537-544, 2003.

2. Hutchinson, H., Hansen, H., Roussel, N., Eiderbäck, B., Mackay, W., Westerlund, B., Bederson, B.B., Druin, A., Plaisant, C., Beaudouin-Lafon, M., Conversy, S., Evans, H. "Technology probes: inspiring design for and with families." In *Proceedings of CHI '03,* pp. 17-24, 2003.

3. Cadiz, J., Venolia, G., Jancke, G., Gupta, A. "Designing and deploying an information awareness interface". In *Proceedings of CSCW* '02, pp. 314-323, 2002.

4. Chewar, C. M., Bachetti, E., McCrickard, D.S., Booker, J. "Automating a Design Reuse Facility with Critical Parameters: Lessons Learned in Developing the LINK-UP System." In *Proceedings of CADUI '04*, 2004.

5. Matthews, T., Dey, A.K., Mankoff, J., Carter, S., Rattenbury, T. "A Toolkit for Managing User Attention in Peripheral Displays." In *Proceedings of UIST '04*, pp. 247-256, 2004.

6. Microsoft. Longhorn. Home page: http://msdn.microsoft.com/Longhorn/
Designing for Longhorn: http://msdn.microsoft.com/Longhorn/default.aspx?pull=/library/en-us/dnaero/html/wux_topic_sidebar.asp

7. Miller , T. and Stasko, J. "InfoCanvas: A highly personalized, elegant awareness display," In *Supporting Elegant Peripheral Awareness, workshop at CHI '03.* 2003.

8. Mynatt, E.D., Back, M., Want, R., Baer, M., Ellis, J.B. "Designing audio aura." In *Proceedings of CHI '98*, pp. 566-573, 1998.

9. Pedersen, E. R., and Sokoler, T. "AROMA: Abstract representation of presence supporting mutual awareness." In *Proceedings of CHI '97,* pp. 51-58, 1997.

10. Tan, D. and Czerwinski, M. "Information Voyeurism: Social Impact of Physically Large Displays on Information Privacy." In *Proceedings of CHI '03*, pp. 748-749, 2003.

11. Weiser, M. and Brown, J.S. "Designing calm technology," *PowerGrid Journal*, **1**(1), July 1996.

Appendix A: Literature survey materials

1. List of peripheral display projects and researchers: http://kettle.cs.berkeley.edu/ambient/11

Appendix B: Interview Questions

Background

1. How did you get the idea for your system? Did you speak with any users before building your system? If so, with whom did you speak, and what did you learn?

2. What is the process of creating a peripheral display from pre-conception to design, to implementation, starting with how you got the idea for your system?

3. What were the main principles you kept in mind during the design and implementation phase? For example, minimizing disruption and how to accomplish this goal.

Design

1. What was the first step in creating any of your peripheral displays? What comes next?

2. What was the hardest part about designing your displays?

3. What were your design decisions regarding human attention based upon? For example, how did you ensure that your display was peripheral/not too disruptive? How did you or do you plan to validate these decisions?

4. Did you consider including alerts on any of your displays? For example, attempting to attract the user's attention when important data arrives. Why or why not?

Implementation

5. What was the first step in implementing any of your peripheral displays? What comes next?

6. What was the hardest part about implementing your displays?

7. Please draw a diagram of how the software works. (Feel free to use another sheet of paper.) Are the different components self sufficient, or do they depend on one another?

8. What parts of the implementation are required specifically because your displays were peripheral? In other words, other than getting some input data and simply displaying it on the output. Is there any significant or difficult code between these two steps?

9. Have you ever used existing toolkits, such as Phidgets, iStuff, or the Context Toolkit? What is your experience using these?

10. If you were to build another similar peripheral display, which code components from your existing displays could you reuse?

11. What programming language(s) did you use? Why?

12. How long were they deployed and used? What caused them to be taken down or to stop working?

Input

13. Is there any complex input processing or analysis you've wanted to do for a peripheral display that you didn't because it would take too long? Examples (video, audio, text, or numerical processing or analysis)?

14. Have you used any other toolkits or third-party code for processing or analyzing input data? How do you currently analyze input?

Interruptibility

15. In any of your displays, have you considered taking a user's interruptibility into consideration when displaying output?

User Customization

16. Did any of your displays allow users to customize the input data or the way in which data is displayed? How (what data was customizable, how did the user do it on the UI, how did you support it in code)?

Evaluation

17. What tests did you run with users? What did the users find that worked well? What problems did the users find?

18. At what stages did you want to evaluate and what metrics would you use at these stages?

19. What do you want to get from your evaluations? What metrics would be important in evaluating your peripheral displays?

20. What would you have done differently in building your system? (i.e. user interactions or design)

Difficulties

21. What was most frustrating about the development process? And what was the must frustrating about the design process?

22. What types of unexpected problems did you run into during implementation?

23. What problems were created due to the hardware you chose?

24. Do you have any specific requests or suggestions for functionality that a toolkit could provide?