Designing for Limited Attention

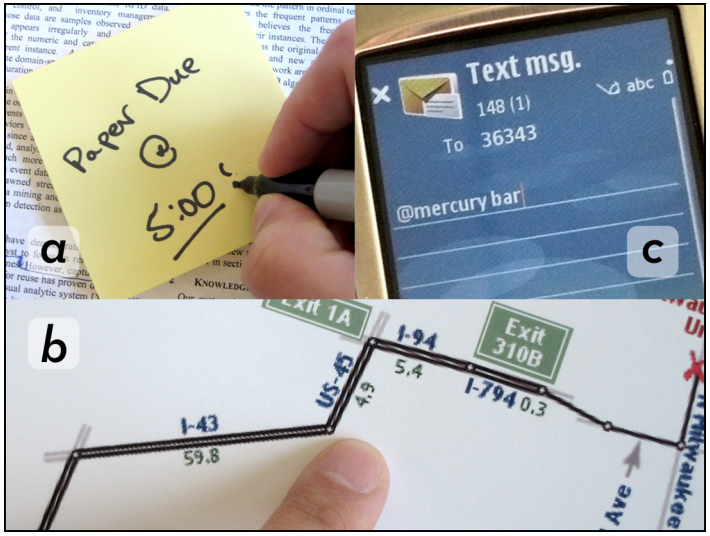

Figure 1. Three interactions designed for limited attention: *a)* jotting a reminder on a sticky note to reduce memory load, *b)* accessing information on a printed route map while driving, and *c)* micro-coordinating an evening's activities using text messaging.

Joel Brandt
Stanford University HCI Group
353 Serra Mall
Stanford, CA 94305
jbrandt@cs.stanford.edu

Noah Weiss
Stanford University HCI Group
353 Serra Mall
Stanford, CA 94305
noahweiss@stanford.edu

Scott R. Klemmer
Stanford University HCI Group
353 Serra Mall
Stanford, CA 94305
srk@cs.stanford.edu

Abstract
Not all time is created equal: in the course of a day, the resources at hand vary dramatically. Activities often span multiple differing use contexts—riding a train, talking with a colleague, attending a meeting, engaging in focused work at the desktop—and these different contexts imply different constraints on action. To work effectively within these constraints, people often divide tasks into multiple phases. For example, jotting a reminder of a future task has little intrinsic value; it serves to distribute one's cognition in service of structuring future action. Similarly, the value of preparing a route map lies in its affordance for rapid consultation while traveling. This case study draws on interviews with developers and on our own research to present considerations for designing interactions spanning times of varying attention.

Keywords
attention, mobile computing, distributed cognition

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces—*Interaction styles, graphical user interfaces, user-centered design*.

Introduction
Significant differences in productivity arise from how one divides his or her time. Choosing how to divide one's limited attention among multiple tasks can be difficult because the perceived value of time and attention varies—a unit of attention at home on a weekend afternoon may

Figure 2. Interfaces for the two interactions involved in completing a diary entry in the 4l8r system. The user submits a snippet using a mobile phone *in situ*, and later completes the full entry at a convenient time over the web.

have a very different perceived value than a unit of attention while bicycling in heavy traffic. On the other hand, the value of completing a task is also not constant —determining whether one is out of milk is more valuable when at the store than after returning home.

As computing becomes pervasive, it is increasingly important for designers to consider users' available attention. This case study discusses seven applications that explicitly address users' shifting attentional resources. We focus on two concerns: designing for limited attention, and "bridging" situations of limited and more plentiful attention. Of the seven applications we discuss, professionals in industry designed five, and the authors designed two. For the five external applications, we conducted semi-structured interviews with the designers. Each interview lasted approximately 45 minutes; four were phone interviews and one was in person. The interviews were structured around six topics: identification of the need, ideation and design of the interaction, testing methodologies employed, difficulties encountered throughout the design process, interesting use cases observed, and issues learned since deployment.

In this case study, we first briefly describe these seven applications. We then share five broad design considerations that emerged. To aid our discussion, we introduce the term *tasklet* to describe a small portion of an activity undertaken in situations characterized by limited available attention.

Overview of applications surveyed

The authors created the *4l8r* system to conduct mobile diary studies [2]. It divides the traditionally atomic task of entry capture into two phases. In the first phase, the user initiates an entry by capturing a small, salient *snippet* of data with their phone by sending a text, picture, or voicemail message to a server. This snippet later serves as a prompt for completing the larger task of recording a full entry. At a convenient time, participants use a web-based interface to expand their snippets into thorough entries.

ButterflyNet, a system developed in our research group, allows users to review, manage, and share a digital version of their paper notebook [6, 9]. The *Idea-Snippets* feature allows users to capturing ideas by submitting small bits of media (text, pictures, or audio) to their digital notebook using a mobile phone. This media shows up in a special section of their digital notebook for later reference.

OneNote is Microsoft's note-taking and management software. Its *side note* feature supports the tasklet of capturing a note. OneNote places these side notes into an "unfiled" section of the user's notebook for later processing.

Google Calendar is an online calendar service that has several features for completing calendaring tasklets while mobile. Users can send a text message (SMS) to add events to their calendar, as well as retrieve either a complete day's schedule or details regarding the next event on their calendar.

BillMonk is a web-based application for keeping track of small debts among a social group, such as those that might arise when paying for dinner. BillMonk also offers a text-message interface which supports the tasklets of recording and settling a debt.

Facebook is a social networking web site. Facebook's mobile tools allow users to complete a variety of tasklets related to managing and communicating with their social group. For example, users can retrieve contact information for individuals currently in their network and add new individuals to their network. Additionally, users can send messages to their social group, "poke" other users, and set their "status" (similar to an away message in many instant messaging applications). Fi-

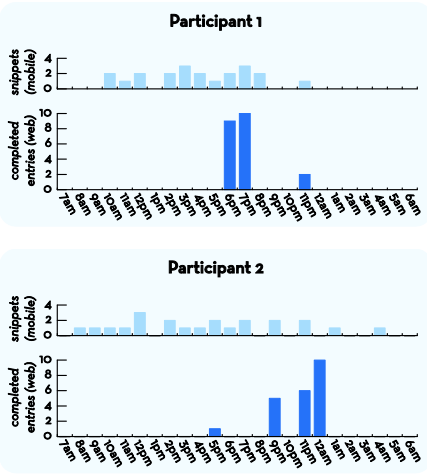

Figure 3. These two pairs of graphs show the usage patterns of two individuals participating in a study ran using 4l8r. For each participant, the total number of *snippets created* and *entries completed* is aggregated by the hour of the day that the interaction took place. These participants completed brief mobile tasklets of submitting snippets throughout the course of their day, and delayed the task of completing full diary entries until more convenient times in the evening.

nally, users can choose to receive updates (messages and pokes that other users initiate) via text messaging.

Dodgeball is a social networking service primarily designed for coordinating impromptu meetings when users are out in social venues such as restaurants and nightclubs. Prior to mobile use, users complete the task of setting up their social network online. A mobile interface allows a user to complete the tasklet of updating her current location to facilitate impromptu meetings. For example, when a user "checks in" at a location, her friends and her friends' friends that have "checked in" at nearby locations are automatically notified of her presence.

1. Supporting variation in the "value" of time

Alex, a project manager for Microsoft's OneNote, explained that OneNote's side note feature was designed for times when the user "has something that comes to mind out of current context and needs a way to record it easily." He went on to say that "a typical scenario we wanted to support was having people use OneNote during a phone call." The side note feature was designed so that "the user doesn't have to *do* anything with the note… users want to 'let me not worry about it until I have to'."

The approach that Alex identifies of explicitly designing for shifting situational constraints – and partitioning interactions across situational constraints – is something of a departure from traditional desktop user interface design. The motivation behind4l8r: we were interested in encouraging participants to document ideas as they struck them – often while mobile. This was problematic, as the times we were most interested in getting data about were exactly those when participants were least able to spare the time to document it. This led us to 4l8r's two-part approach of having participants create a snippet in the field, followed by a fuller report over the web. Figure 3 presents the mobile and web usage pattern of two representative participants. Note that both submit snippets from their phone throughout the day, and both have clear preferences for completing full follow-up entries in the evening.

In short, providing explicit support for splitting apart tasks – *e.g.* the recording of a diary entry or the planning and coordination of an evening out – can provide significant user experience benefits. This partitioning can provide facilities for distributing cognition [3] and offering users more control over their time.

2. Making information available and visible at the "right" time

The need to store and retrieve information often arises in situations where time is highly valued. Distributing cognition – *e.g.*, through a written reminder – is time sensitive, as one might forget. When an idea comes to mind, for example, significant value is gained in immediately instantiating that idea physically, for example, by writing it on a sticky note. Similarly, in the case of retrieving information – such as the next event on one's calendar – it is often most valuable when obtained quickly.

Alex observed that in OneNote, "people leave [their Side Notes] around on the screen… user's said they wanted to 'make sure I can record it and later go back and process it'." In this way, the side note feature is used for both the tasklet of recording information and the tasklet of quickly accessing it [4]. Alex went on to describe particular design decisions that facilitate these tasklets. "[A side note window] is more of a pad than an individual sticky note." The pad has backward and forward arrows so the user can "riffle through all side notes." Each side note window is a *view* into the same pad, and the user can have as many side note windows open as they like. Additionally, users can "pin down" windows so that they are always on top of other windows. These interface elements extend beyond the functionality of the traditional sticky note that is being metaphorically invoked, providing users with additional control over their information.

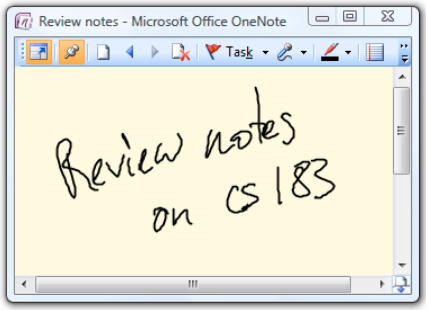

Figure 4. A side note in Microsoft's OneNote. The arrows in the toolbar support "riffling" through the user's stack of side notes. Handwriting is automatically recognized so that it is searchable, *e.g.* this note has automatically been titled "Review notes" based on the handwriting in the note.

As espoused by the popular literature on time management – such as Allen's *Getting Things Done* [1] – filing information according to a "context" in which it will be useful to retrieve it can smooth workflow, limit overload, and increase productivity. We turn, then, to the question of how digital tools can efficiently support this "filing" step. While OneNote provides extensive features for manually filing one's notes, filing is rarely completed during high-value time. As such, all side notes end up in a special "unfiled" section of the user's notebook so that they may be easily filed on "cheap" time. Alex notes, however, that "people break down into one of two groups: pilers and filers. Pilers want to just get info down and don't care where it goes. So we made search as good as possible for them."

In our work with the IdeaSnippets feature of ButterflyNet, the importance of filing information in the "right" location was made particularly evident. While we intended for users to use IdeaSnippets as starting points for further ideation, the snippets were filed inside a tool intended primarily for review and sharing. As a result, this feature received lower than expected usage.

Being able to retrieve data at the right time allows people to be more extemporaneous. Carl, a designer for Google Calendar, notes that the "philosophy behind Calendar was to make it extremely easy to add and see events. … A lot of people need some way [to manage their events] but few use electronic calendars … paper wins because you can take it with you." Early in the design process, they "sat down and tried to come up with things [people do with their calendar] while away from their computer." There were "three main things: 'oh shoot, where do I need to go next?', 'I'm waking up in the morning, what do I have going on?' and 'Out to lunch or driving and think *oh shoot, I need to remember to meet with so and so*'". GVENT supports these three tasklets.

3. Understanding the social dynamics of tasklets

When designing interactions for tasklets that involve multiple people, several social factors come in to play. Consider the activity of coordinating a group of people for an evening out. Traditionally, this activity requires significant upfront planning – soliciting suggestions and availability information from the individuals involved, making a decision, confirming that it is acceptable to all parties, and the like. While this approach allows the bulk of work to occur at "cheap" times, it is quite brittle to people's changing schedules and desires. Recent mobile technology has increased the prevalence of *micro-coordination* [5], where individuals replace or augment "planning ahead" with the use of mobile devices to communicate immediately preceding a yet-to-be-determined event. We suggest that while micro-coordination has significant value, it can potentially place a great deal of undue burden on the individuals involved. For example, individuals who are not free (and would have happily declined an invitation during their "cheap" time) may be interrupted by numerous phone calls from other parties. Many new users of mobile phone technology express concerns along these lines, suggesting that they do not want to become a "slave to the phone" [8].

Dodgeball addresses this burden by providing explicit support for micro-coordination. Originally, Dodgeball was built for the designers and their friends as a lightweight way for them to engage in social micro-coordination after being laid off from a dot-com bust. With Dodgeball, users specify the individuals in their social network using a web interface in their "cheap time". Then, when they are out on the town, they "check in" at a location with a simple text message. Other users in their network who have recently checked in at a nearby location are automatically notified of their presence. This enables very low-cost micro-coordination that does not burden uninterested parties.

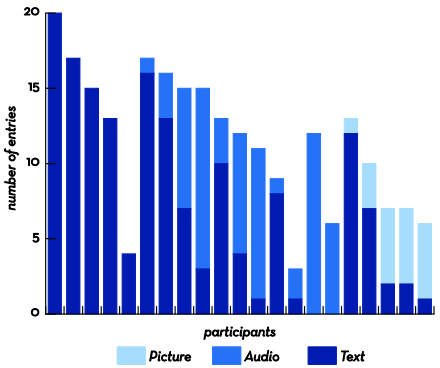

Figure 5. This graph shows number of entries submitted by participants in a study ran using the 4l8r system. The number entries submitted by each participant is shown, further broken down by media type used for each snippet. Participants are clustered based on media types used.

From the perspective of the user "checking in," the bulk of Dodgeball's benefit could be achieved by sending a text message to a distribution list containing all of that user's friends. However, this would result in the other users receiving messages *regardless* of whether they are nearby, and most users – spammers aside, of course – are sensitive to whether a recipient of their message will find it a nuisance. The Dodgeball designers further recognized that users appear more willing to shoulder a burden that saves others time when that burden can occur on "cheap" time.

Similar social issues arose when designing the feature of Facebook that allows users to receive messages and "pokes" via text messaging. Mark, the lead designer of Facebook's mobile services, stated that a significant amount of thought was given to the issue of interruption. Regarding this issue, Mark said "If I'm sending you a message on Facebook [using the web site], does it change whether or not I want to send you the message if I know it's going to go to your phone? … I have to think about if I want to maybe interrupt you." Putting this control in the hands of the senders, he felt, might relax this particular tension, but would take away power from those users opting in to the service. Mark expressed particular concern over situations where a user would expect to receive a message on her mobile device after opting in to the service, only to discover that she had missed it (*i.e.* the message could only be accessed using the web site) because the sender chose not to interrupt her. Ultimately, Facebook decided that complete control should be given to the recipients.

Privacy issues are a common concern in social software, and introducing tasklets that divide interactions across time can result in users paying little attention to these issues until it's too late. "People don't think about the consequences of having 500 friends on Dodgeball" when they're setting up their social groups online, says Dennis. "It isn't like Friendster (another social networking website) where it's no big deal… Dodgeball suffers from a severe 'ex-girlfriend bug'", meaning that unless one takes care to block an ex-significant-other, for example, he or she will receive a notification each time the user checks in.

4. Determining the right balance between functionality and complexity

In designing any interaction, there exist tradeoffs between flexibility, functionality, complexity, and understandability. Our goal here is not to address all of the factors which influence these trade-offs, but instead to bring up the interesting factors that arise when developing interactions for limited attention situations.

In general, it is valuable to offer multiple modalities of input when designing interactions for tasklets. This is particularly true for tasklets involved in distributing cognition. During our work with 4l8r, for example, we found that flexibility of input was very important, but for reasons we didn't expect. Our original hypothesis was that participants would choose among the various media available for snippet reporting based on the context they were in. For example, when users were in a noisy restaurant, they might choose to use text over audio. We found instead that media choice was largely a personal preference. Figure 5 shows the breakdown in media choice for each participant. In the exit interviews, participants often felt strongly about the media type they preferred. S16 said, "Text was easier and faster. I only had to write one or two words to remind myself." S23, on the other hand, hated text: "Voice was so much easier. I hate the T9 and I'm slow with the original text." One participant felt that text was often the most appropriate *media* for snippet reporting, but strongly disliked typing on the phone. Instead, she often chose to handwrite her snippet on a readily available piece of paper, and then take a picture of the snippet and submit the picture via multimedia messaging.

Flexibility of input was a paramount goal when designing the side note feature of OneNote, Alex reported. As a result, input can come from the keyboard, from a stylus when using a tablet, from the screen itself by taking a screen grab, or from a microphone or webcam. This flexibility is valuable when using side note for distributed cognition tasklets. However, because the input is multimodal and unstructured, this creates great difficulties in automatically organizing and indexing the data. The designers of the side note feature wanted to provide enough automation such that "it's still useful if the user doesn't do anything" after taking the note – they "can still use it in searching." As a result, they underwent the arduous task of implementing character recognition support for both handwriting and screen grabs as well as speech recognition for audio and video in order to make notes searchable. This underscores the point that flexibility of input can often be easy to implement for the tasklet interaction, but can make integrating the tasklet into the larger activity quite difficult.

This also brings up the issue of automation. As we mentioned earlier, significant value is often gained by getting information to the "right" place as soon as possible. A seemingly convenient way to do this when designing tasklet interactions for distributed cognition is to introduce automatic processing of submitted data. Deciding on the right level of automation, however, is an important issue. While the side note feature supports automatic indexing to enable search, Alex notes that "we didn't try to design too much into the experience after you take a [side] note … there are just too many different styles [of user behavior]." During early trials "testers said 'it would be nice if you could help me file or auto file the side notes'." Ultimately, however, they determined that this level of automation was not a priority, and that for "version one we should create something that everyone would understand and was really simple." The side note feature has been in use for four years, and they "simply haven't seen a need to go to version two."

Carl also reflected on the issue of automation in the design of Google Calendar. He stated that one goal of Google Calendar was to "make it extremely easy to add" events by "removing forms from the experience" – that is, they wanted to make adding events to a digital calendar a tasklet. Out of this grew the *quick add* feature, which lets users enter events into their calendar using natural language like 'dinner with tracy at nobu, 9pm on 2/17'. "GVENT leverages the quick add feature and makes it accessible while users are away from their computer." Allowing users to immediately get information to the right place insures that information will be accessible at the right time.

Carl reported that too much automation led to problems in early versions of the quick add feature. Initially, they had "a smarter version of quick add [and it was] universally hated. For example, if I entered 'dinner with tracy at nobu', quick add would parse this, look up 'tracy' as a person and automatically send an invite, and put 'nobu' in the location. So, all that would end up in the description was 'dinner'. People felt that the important points didn't end up in the description. The result was that people wanted far *less* automatic processing."

Furthermore, automation often introduces a need for structuring input, which can be a barrier for distributing cognition in an expedient manner. For example, using GVENT, a user is forced to construct input that contains all of the details of an event, rather than just submitting 'dinner 9', and at a more convenient time using this as a prompt to enter the full details of the event. The designers of GVENT considered the trade-off between freeform input and having to construct an entry that provided enough detail for automatically creating the entry. "There's a difference between e-mailing yourself and making an entry on your calendar." Carl noted. "We

thought of trying to store and present unstructured events and having users deal with events later" but "we aren't a note-taking product, and didn't want to clutter the calendar interface with things to refine freeform text." Carl jollily observed that there are other products that do a great job dealing with freeform text while mobile – "notably the mobile Gmail client" – and stated that "user should use that if that's what they want."

As Norman points out, there are a host of other problems associated with automation [7]. Central to our discussion is the fact that when automation fails during a tasklet, the user may not be in a position to fix the problem – that is, they may not have the time or tools necessary to correct the error. While we defer the majority of our discussion on feedback to the following section, we suggest that when automation is involved, there should at the very least be sufficient feedback to let the user know whether or not the tasklet was completed as intended.

The issue of requiring semi-structured or structured input brings up the important issue of the level of complexity users are willing to tolerate in tasklet interactions. Dennis initially gave a tongue-in-cheek response when asked how the set of mobile features was chosen: "we maxed out at three commands because that's all of the easily accessible symbols on a cell phone keypad (@, !, ?)". He went on to say, however, that it was really "a question of how much people can remember. You tell them 3, and they say 'ok', you tell them 5 and people get lost." The designers of BillMonk confirmed this problem. Initially, they supported a wide variety of tasklets, and distributed business cards that summarized the associated commands. They quickly learned, however, that referencing an external resource in order to complete a tasklet is typically too high of a barrier. We suggest that there are three "plateaus" of complexity. First, there is the level at which a user can completely internalize the operations – here, roughly three commands. Then, there is the level at which a user can remember the complexity, must expend some mental energy to remember it – here, perhaps five commands. Finally, there is the level which requires external resources to remember the functionality. We suggest that for typical tasklets, users are unwilling to go beyond the first plateau of complexity.

Complexity of the individual commands is obviously complicit in this discussion, and is an issue that must be addressed when designing tasklet interactions. For example, Dodgeball has a "command line interface" where users send commands to the service via text messaging. Originally, this interface had a rigid syntax of *command modifier*. Dennis notes that "this made sense for '@ *location name*' [to check in at a location], but it doesn't make sense for '? *location name*' [to find out where a place is]. We got a lot of things like 'mercury bar?' instead of '? mercury bar'. People didn't get it because it wasn't like English."

5. Providing appropriate levels of feedback

Appropriate feedback is a central goal in any interaction design. As such, it is again our goal not to focus on the entire issue of feedback, but instead discuss the specific points that are unique to the design of tasklet interaction.

Users are hesitant to adopt systems that don't offer clear and immediate feedback. However, feedback during tasklets can be a burden due to the time and attention they require. This both means that feedback is important for "important" tasklets and a nuisance for "minor" tasklets.

As was alluded to earlier, due to the automatic parsing that was involved, feedback was seen as a very important issue in the design of Google Calendar's GVENT feature. "Any time we're doing this magical parsing, we need to provide explicit feedback that doesn't interrupt the flow but tells them what we did." Carl noted. By sending them back an SMS telling them what action was

performed "they can later go back and make a correction using the response if there was an error." But because the parsing works correctly most of the time it's "also a confidence building thing – people can make sure they did it right." We suggest that for "important" tasklets, it is important to provide feedback that both confirms a successful operation and provides enough information to correct a failed operation. This means that in the case of GVENT, it would not be enough to provide feedback of the forms "event successfully added" and "could not understand your entry." The former does not convince the user that the parsing was *actually* correct, and the latter does not remind the user of what they submitted so that they can attempt to fix the problem.

Interestingly, Carl notes that feedback was not initially part of the quick add feature on the desktop, but they "got negative feedback" because users weren't sure if it worked. So, they "did it right from the beginning" with GVENT.

Because Facebook's various interactions that support tasklets while mobile span a range of "importance", the designers opted to give users control over what events generate feedback. For the very pragmatic tasklets – looking up contact information or adding a new individual to one's network – feedback is always provided. However, for tasklets which are less important, such as "poking" someone, users can choose not to have feedback sent. Users opt out of this feedback when it has a higher cost than the benefit it provides. It is important to point out that in this instance, the cost is largely monetary – receiving an SMS costs money. However, even without this issue, the feedback consumes time, which may be a precious resource when completing a tasklet.

When designing both 4l8r and the IdeaSnippets feature of ButterflyNet, we decided not to provide any feedback to users during snippet submission. Interestingly, no participants in our evaluation of 4l8r complained about this, but several users of ButterflyNet did. We suggest that this is difference is likely due to the fact that the snippets in 4l8r held little value to the submitters – they were simply a tool to help provide more data for the study they were participating in.

Conclusions

We believe obtaining a deeper understanding of designing interactions for limited attention to be an important step on the path to pervasive computing. This case study has provided a set of issues which we believe designers should be attentive to when designing these interactions.

References

[1] Allen, D., Getting Things Done: The Art of Stress-Free Productivity. London: Penguin Group 2001.

[2] Brandt, J., N. Weiss, and S. R. Klemmer. txt 4 l8r: Lowering the Burden for Diary Studies Under Mobile Conditions. In Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems. ACM Press, 2007.

[3] Hollan, J., E. Hutchins, and D. Kirsh. Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research. ACM Transactions on Computer-Human Interaction 7(2): ACM Press New York, NY, USA. pp. 174-96, 2000.

[4] Kirsh, D. The Context of Work. Human-Computer Interaction 16: Lawrence Earlbaum Associates. pp. 305-22, 2001.

[5] Ling, R. We Will Be Reached: The Use of Mobile Telephony among Norwegian Youth. Information Technology & People 13(2). pp. 102-20, 2000.

[6] Maldonado, H., B. Lee, and S. R. Klemmer. Technology for Design Education: A Case Study. In Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems. ACM Press. pp. 1067-72, 2006.

[7] Norman, D. A. The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation'. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 327(1241): The Royal Society. pp. 585-93, 1990.

[8] Palen, L., M. Salzman, and E. Youngs. Going Wireless: Behavior & Practice of New Mobile Phone Users. In Proceedings of ACM Conference on Computer Supported Cooperative Work. ACM Press. pp. 201-10, 2000.

[9] Yeh, R., C. Liao, et al. ButterflyNet: A Mobile Capture and Access System for Field Biology Research. In Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems. ACM Press. pp. 571-80, 2006.