Wait-Learning: Leveraging Wait Time for Education

Carrie J. Cai
MIT CSAIL
Cambridge, MA
cjcai@mit.edu

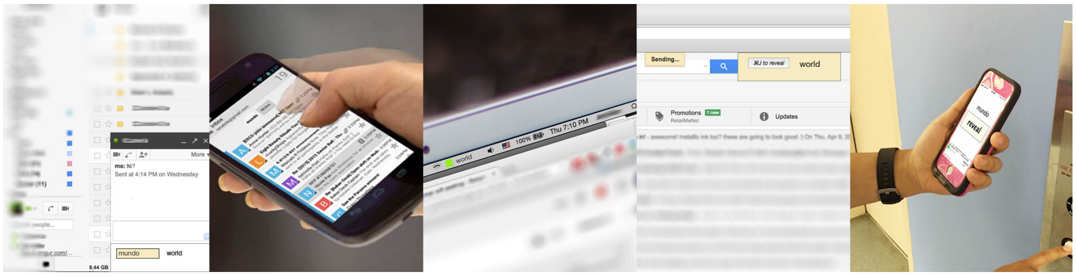

Figure 1. Wait-learning enables learning during micro-waiting moments. WaitChatter and WaitSuite enable learning across a variety of task contexts: (left to right) instant messaging, pull-to-refresh, wifi seeking, email sending, and elevator waiting.

ABSTRACT

Competing priorities in daily life make it difficult for those with a casual interest in learning to set aside time for regular practice. Yet, learning often requires significant time and effort, with repeated exposures to learning material on a recurring basis. Despite the struggle to find time for learning, there are numerous times in a day that are wasted due to micro-waiting. In my research, I develop systems for *wait-learning*, leveraging wait time for education. Combining wait time with productive work opens up a new class of software systems that overcomes the problem of limited time while addressing the frustration often associated with waiting. My research tackles several challenges in learning and task management, such as identifying which waiting moments to leverage; how to encourage learning unobtrusively; how to integrate learning across a diversity of waiting moments; and how to extend wait-learning to more complex domains. In the development process, I hope to understand how to manage these waiting moments, and describe essential design principles for wait-learning systems.

Author Keywords

wait-learning; micro-learning; attention management

UIST '15 Adjunct, November 08-11, 2015, Charlotte, NC, USA
ACM 978-1-4503-3780-9/15/11.
http://dx.doi.org/10.1145/2815585.2815589

ACM Classification Keywords

H.5.2. User interfaces: Interaction styles

INTRODUCTION

There are numerous times in a day when people wait, such as time spent waiting for an elevator, waiting for an instant message reply, or waiting for a lecture to start. Yet, these moments are rarely used for productive tasks due to the perception that long, uninterrupted periods of effort are required to make meaningful progress. At the same time, spaced, repeated exposure [3] to educational content is key for endeavors such as learning a foreign language or studying for a standardized exam. While people may have the desire or the *choice motivation* to form productive habits, many lack the *executive motivation* to actually perform these tasks on a repeated basis [4], due to barriers in finding time for practice. Given existing tendencies to fill wait time with less productive activities (e.g. compulsive email-checking), there is an opportunity for learning to occur during these daily gaps that may otherwise be spent unproductively.

This work presents *wait-learning*: leveraging wait time for education. I design and build interactive systems for learning during waiting moments, bringing in expertise from domains such as attention management, education, and task allocation. For example, **WaitChatter** delivers learning exercises while users wait for instant message (IM) replies, and **WaitSuite** (Figure 1) integrates learning across five different modalities and waiting contexts. Although user interaction should be seamless, the underlying systems may be complex in determining the optimal moment to intervene, and the kinds of learning exercises to deliver at that moment to maximize learning without sacrificing engagement.

Through my research, I aim to address the following research questions:

- Which waiting moments are more compatible with learning, given factors such as expected wait time and cognitive load?
- How do we design interfaces to encourage engagement in learning while minimizing intrusiveness?
- How can wait-learning serve as an entryway to continued learning or productive work, helping people ease in to more complex activities during longer periods of wait time?

THESIS STATEMENT

Wait-learning is a novel and effective way to learn, increases engagement in learning, reduces frustration in wasted time, and minimizes the perception that time spent learning is intrusive to daily life.

RELATED WORK

There is a growing body of work aiming to integrate micro-learning into existing daily activities, ranging from second language learning of location-relevant vocabulary [5] to displaying parts of a web article being read into a foreign language [12]. A common goal of this prior work is to lower the activation barrier of performing micro-tasks. However, prior work has tended to focus more on the *content* delivered rather than the *timing* of those deliveries. Moreover, these systems often leveraged *live* time, moments when the user is already engaged with an existing task, rather than *wait* time.

Micro-Learning and Micro-Work

Based on evidence that spaced and repeated exposure [3] to educational content results in greater learning gains, a rich thread of research on *micro-learning* [6] has explored methods to distribute learning into small units throughout a person's day-to-day life. Micro-learning has largely been implemented in the form of mobile applications that provide opportunities to learn contextually relevant second language vocabulary based on surrounding contexts [2,5]. Several desktop applications have also embedded learning programs into common daily activities, such as Lernschoner [6], in which learners complete a vocabulary flashcard before resuming activity on their computer screen, and Aloe, where some words within articles the user is browsing are displayed in a foreign language [12].

Despite the promising benefits of micro-learning, researchers found that some users discontinued usage due to the slower rate at which they could conduct their normal tasks on the Web [12]. Another also found that the quantity of vocabulary reviewed may depend as much on when users were available to learn as the contextual relevance of those vocabulary [5]. Challenges raised in prior work point to the necessity of identifying moments that are non-intrusive to a user's daily workflow, as even micro-diversions may be viewed as disruptive if the user feels that his or her tasks are being delayed. Motivated by these challenges, this work targets moments when users would ordinarily spend waiting, with the goal of minimizing perceived disruption of ongoing tasks. Because filled waiting periods are typically perceived to be shorter than unfilled waiting periods [8], wait-learning could also enhance a user's subjective experience beyond providing educational benefits.

Attention and Interruption

Although wait-learning aims to leverage idle time, previous work on interruptability suggests that the detailed manner and timing of delivery can still result in significant differences in task performance and levels of frustration. Because our minds dynamically allocate and release resources throughout task execution, the timing of information delivery relative to a user's ongoing task may affect interruption cost [7]. Since decreases in workload tend to be larger at boundaries corresponding to larger chunks of a task [7,9], a system that presents information during down time should favor boundaries that represent more salient breaks in workflow [7]. My research aims to understand how to design wait-learning experiences in a way that minimizes disruption.

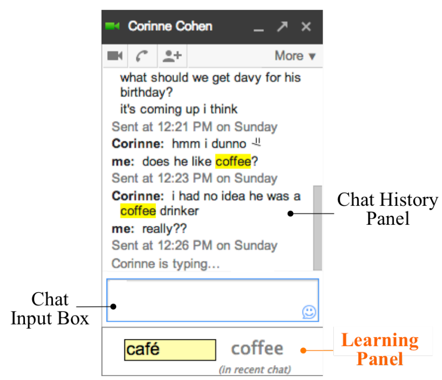

Figure 2. WaitChatter presents vocabulary exercises in the *learning panel* while the user awaits an IM response. Here, the user is being quizzed on a word and must enter the second language (French) translation given the native language (English) prompt. The word is highlighted in the chat history because it appeared in the context of the conversation.

WAITCHATTER

As a first instantiation of wait-learning, I developed an extension of instant messaging (IM) called WaitChatter [1] (Figure 2). WaitChatter presents second-language vocabulary exercises while the user awaits an IM response. Instant messaging offers a powerful opportunity for wait-learning due to its semi-asynchronous nature. Because messages being typed are unseen by the conversant, a user must often wait in anticipation of a response, with no guarantee that the other party with in fact reply [11].

In WaitChatter, users see vocabulary quizzes appear under their chatbox while waiting for an IM response, and can optionally complete these exercises while waiting. Because vocabulary retention hinges on repeated exposure to words across time, short spurts of waiting during instant messaging may be a low-barrier way to get repeated practice.

Implementation

WaitChatter automatically detects when a user might be waiting for a response in two ways. The first case occurs after a user has sent a chat message and is waiting to see whether his conversant will respond. The second case occurs when the conversant has started typing but not yet sent the message. WaitChatter triggers an exercise when the "[conversant] is typing…" indicator appears in the user's chat window and if the user if not still typing.

WaitChatter uses the Leitner flashcard algorithm to dynamically schedule the order of vocabulary exercises: repetitions occur at increasingly spaced intervals so that they are encountered just as they are about to be forgotten. WaitChatter extends the Leitner algorithm in two ways. First, aside from teaching learners commonly used words, WaitChatter also selects words from the IM conversation on-the-fly and automatically converts them into quizzes for just-in-time learning. In order to dynamically interleave between contextual and non-contextual vocabulary, WaitChatter modifies the Leitner algorithm to allow for timely injections of contextual vocabulary. Second, WaitChatter also modifies the algorithm to increase engagement by dynamically reordering words based on whether the user has previously ignored the exercise.

Engagement and Learning Value

My research on WaitChatter found that users were fastest to engage and most likely to engage with learning when exercises were delivered at the start of the waiting period, compared to other timing conditions. Results also showed that learning during wait time can serve as a viable channel for learning, at least for bite-sized information. After two weeks of casual WaitChatter usage during their regular IM activities, users on average retained 57 new Spanish and French words, the equivalent of learning approximately four new words per day.

WaitChatter's primary value to learners is the decreased effort involved in making time for learning. WaitChatter users cited time and ease of access as a major benefit, contrasting WaitChatter to mobile applications which required them to consciously set aside time and open the app to learn. Given the critical importance of time, I am continuing to examine how wait-learning systems can accommodate a wider range of waiting moments with varying amounts of wait time.

Live Deployment

After the initial research study, WaitChatter was deployed to the public as a Google Chat extension and has since been widely circulated, with over 900 installations. The substantial public interest in WaitChatter validates the need for learning despite limited time. At the same time, the abundance of people who voiced interest in wait-learning, but who could not use WaitChatter because they do not instant message, motivates a system for integrating learning into a diversity of waiting moments.

WAITSUITE

In many educational disciplines, the key to retention is the ability to sustain repeated exposure to the same content over time. However, many learners only experience certain types of waiting on particular days, or may encounter unusual periods with limited access to particular platforms. To address this problem, I developed WaitSuite, a system that unifies a suite of wait-learning possibilities into a single, integrated learning experience. WaitSuite unifies learning across the following five apps: 1) **ElevatorLearner:** users learn while waiting for the elevator. 2) **PullLearner**: users learn after pulling to refresh mobile app content, while waiting for the content to load. 3) **WifiLearner**: users learn while waiting for their computer to connect to wifi. 4) **EmailLearner**: users learn after sending an email, a time when s/he has potentially just completed one task before transitioning to something new. 5) **ChatLearner**: users learn while awaiting IM responses ChatLearner is essentially the same as WaitChatter, but without contextual words.

Because different kinds of waiting can occur at different frequencies, the ability to maintain forward progress *across* apps is key. WaitSuite synchronizes data so that users can continue to make learning progress on the same set of vocabulary when switching from one app to another. For apps like WifiLearner which specifically target internet down time, handling offline usage is a necessity. WaitSuite caches data on the client side so that the user can continue to complete exercises while offline. Because apps targeting internet down time may fetch consistently stale exercises, WaitSuite also force-synchronizes to the server every time internet re-connects. In the case of a concurrency conflict, WaitSuite selects the state reflecting further progress.

DESIGN SPACE

Given the vast diversity of waiting moments, it is necessary to examine trade-offs between different kinds of waiting, and determine which ones are more suitable for wait-learning. I charted a design space that spans three critical dimensions: 1) wait time, 2) response time (time to begin responding to the learning exercise), and 3) cognitive and perceptual load (the likelihood of other processes competing for the user's attention).

Using the five instances of wait-learning described in WaitSuite, I gathered in-situ data on real usage patterns across apps to determine metrics such as wait time, response time, and engagement rate. I found that engagement rate is highest when typical response time is less than the wait time, and when cognitive load is low. I also found that wait-learning can reduce the frustration associated with certain kinds of waiting (i.e. waiting for wifi). In evaluating these systems, I am currently establishing design principles that can be used by other designers of wait-learning systems.

WORK IN PROGRESS: SELF-SOURCING

While waiting could conceivably accommodate simple tasks like vocabulary learning or photo labeling, it is more difficult to imagine how large, complex tasks such as learning calculus or writing a project proposal could be feasibly "self-sourced" [10] during down time. Even if large tasks can be broken down into micro-components, it is unclear how individuals can complete those smaller components without context of the bigger picture.

As a next step in my thesis, I plan to expand wait-learning beyond simple tasks, by examining how micro-tasks can serve as an entryway to more complex tasks. For example, a system could order tasks such that lightweight, context-free tasks are presented first. In completing these smaller tasks, the user may gain the contextual awareness necessary to complete the longer task. I am building a self-sourcing system component that automatically orders tasks based on a number of interdependent task parameters.

FUTURE WORK: GROUP WAIT-LEARNING

Waiting occurs not only in personal contexts, but also in group settings, such as in the minutes before a lecture starts, or at the start of a business meeting. In these situations, wait-learning could empower not only the completion of personal tasks, but also the education of groups with a common goal. However, group-waiting also bears a number of challenges, such as different arrival times between individuals and varying expertise. In the upcoming year, I plan to address these challenges. For example, a system could leverage the collective intelligence of the group, using differential arrival time as a way of progressively narrowing down answer options in a group learning exercise so that late arrivers benefit from early arrivers.

SUMMARY

My work thus far has sought to answer critical questions such as which design dimensions are more effective for wait-learning (WaitSuite), which moments maximize engagement within a particular domain (WaitChatter), and how to design the user interface to encourage learning while minimizing disruption (both). It has also demonstrated that wait-learning can be effective for bite-sized learning. In future work, I plan to understand how wait-learning can expand beyond simple, personal learning tasks, by developing systems that involve more complex goals and coordination among groups. At the conclusion of my graduate work, I hope to have built systems that enable users to more easily complete meaningful tasks, demonstrated the efficacy of these systems, and established essential design principles for wait-learning systems.

ACKNOWLEDGMENTS

I am grateful for the support and mentorship of my thesis advisor, Rob Miller, Professor of EECS at MIT CSAIL, as well as Shamsi Iqbal and Jaime Teevan at Microsoft Research. This research was funded by Quanta Computer and MIT Lincoln Laboratory.

REFERENCES

1. Carrie J Cai, Philip J Guo, James Glass, and Robert C Miller. 2015. Wait-learning: leveraging wait time for education. In CHI'15. ACM.
2. David Dearman and Khai Truong. 2012. Evaluating theimplicit acquisition of second language vocabulary using a live wallpaper. In CHI '12. ACM, 1391–1400.
3. Frank N Dempster. 1987. Effects of variable encoding and spaced presentations on vocabulary learning. Journal of Educational Psychology 79, 2 (1987), 162.
4. Zoltan Dornyei and Istv´an Ott´o. 1998. Motivation in action: A process model of L2 motivation. Working Papers in Applied Linguistics (1998), 43–69.
5. Darren Edge, Elly Searle, Kevin Chiu, Jing Zhao, and James A Landay. 2011. MicroMandarin: mobile language learning in context. In CHI '11. ACM, 3169–3178.
6. Gerhard Gassler, Theo Hug, and Christian Glahn. 2004. Integrated Micro Learning–An outline of the basic method and first results. Interactive Computer Aided Learning 4 (2004).
7. Shamsi T Iqbal and Brian P Bailey. 2005. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In CHI'05 Extended Abstracts. ACM, 1489–1492.
8. David H Maister. 1984. The psychology of waiting lines. Harvard Business School.
9. Yoshiro Miyata and Donald A Norman. 1986. Psychological issues in support of multiple activities. User centered system design: New perspectives on human-computer interaction (1986), 265–284.
10. Teevan, Jaime, Daniel J. Liebling, and Walter S. Lasecki. "Selfsourcing personal tasks." *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014.
11. Tree, J. E. F., Mayer, S. A., and Betts, T. E. Grounding in instant messaging. Journal of Educational Computing Research 45 , 4 (2011), 455–475.
12. Andrew Trusty and Khai N Truong. 2011. Augmenting the web for second language vocabulary learning. In CHI '11 . ACM, 3179–3188.
13. Stuart Webb. 2007. The effects of repetition on vocabulary knowledge. Applied Linguistics 28, 1 (2007), 46–65.