Designing Interactive Systems for the Experience of Time

Lassi A. Liikkanen and Paula Gómez Gómez
Helsinki Institute for Information Technology HIIT, Aalto University
P.O.Box 15600, FI-00076 Aalto, Finland
firstname.lastname@hiit.fi

ABSTRACT
User experience (UX) has become a central theme in user-centered design. It draws attention to the experiential side of use and sets new requirements for software design. Waiting and interruptions can be a source of negative experiences with interactive systems. This review paper discusses the subjective experience of time as a part of UX. It also provides advice on how to influence UX through time-considerate design. We start from the psychological theory of time perception and expand this cognitivist model to a direction compatible with affective computing. The outcome describes the interplay between cognition and affect through a concept of subjectively experienced time. We introduce six design guidelines for time-considerate design of everyday applications. Our design implications concern dialogues, progress indicators, and cost-effective system optimization relevant for UX. We point out the need for empirical research to fill in the gaps in the literature, particularly regarding user adjustment and its implications.

Author Keywords
HCI design and evaluation methods; interaction design theory, concepts and paradigms; psychology; user experience; user models.

General Terms
Human factors; design; measurement.

INTRODUCTION
The best things in life seem to happen in an instant. Losing track of time is one of the properties of a “flow” experience, which is deemed highly satisfying, challenging, and optimally engaging [10]. In contrast, waiting can evoke quite the opposite emotions. Emotions influence our behavior and thinking. For instance, when reflecting on past life we tend to remember highly emotional things better than neutral events [3, 4]. When recapping personal experiences, we take their emotional peaks and final moments as representative of the whole [17]. In personal computing, momentary delays and malfunctioning seem commonplace. Consequently our memories of interactions with computers can be biased by these negative experiences, often related to waiting [37]. What if designers could influence these experiences? In this review paper we connect the concept of user experience (UX) to the experience of time and provide evidence and suggestions on how interaction design could modulate them.

User experiences occur and are recalled over time. One recent account of experiences with technology emphasizes the continuous sensory connection with our environment situated in time that creates felt life [40], the ultimate UX. In contrast, the prevailing cognitivist history of HCI research thus far has typically reduced time to system response and user response times [11], which seem outdated and incompatible with recent UX debates. The lack of design perspectives has been noted, but neither fulfilled [29] nor recently updated [see 6]. We also think that the psychological understanding of human time experience has not been truly utilized for HCI nor articulated as design implications. This paper tries to fill that gap and argues that by considering subjectively experienced time (SXT) as a user requirement, one can improve the experiences of everyday interactive systems.

In this paper, we introduce central findings and theories from the psychological literature that concerns time perception. The intention is to expand a cognitivist theory of time perception towards an interactional view [30] of UX and emotion. We provide concrete guidelines about how to design and assess systems for time-related interaction issues, particularly those related to waiting. We target designers and software developers working on interaction design and UX, with systems that are used to solve recurring, everyday computing tasks. Our goal is to answer the following questions that are relevant for designing for UX. A) What influences subjectively experienced time? B) How might system optimizations be prioritized for subjectively experienced time? C) How could slow system responses be handled to mitigate the experience of waiting?

Importance of time in designing for user experience
The main premise of this paper is that experiences are evaluated positively when the passing of time is not noticed, or at least when no waiting is perceived. Previous research shows that this is a shared belief, at least in Western cultures [58]. Making people believe the time has flown (advanced quickly) affects their enjoyment of the task – even if this belief is inaccurate (ibid.). To help design such systems, we introduce the concept of subjectively experienced time to account for time.
CONCEPTS RELATED TO THE EXPERIENCE OF TIME

User experience
The importance of time-related factors for software design is most conveniently articulated through the concept of user experience. The measurement, definition, and delivery of UX has been one of the recent trends in HCI and design communities [15, 26, 27]. The international standard ISO 9241-210 [32] defines user experience as a “person’s perceptions and responses resulting from the use and/or anticipated use of a product, system or service.”

While HCI and design communities have embraced UX, theoretical developments in defining what UX is have advanced slowly [36]. Beyond the standard definition, there is little agreement about the exact notion of UX. The shared view seems to be that the emergence of UX [e.g. 27, 33, 56] has drawn attention from functional, behavioral, and rational aspects of use to affective, aesthetic, and phenomenal ones [44]. Many would agree that UX is about user satisfaction beyond usability or problem solving.

UX has also led researchers to redefine and expand the notion of user from a goal-oriented performer towards a more holistic view of a user as a person with feelings and taste [35, 73]. McCarthy and Wright [40] emphasize the embodied, holistic experience of technology use which involves constant meaning making out of interaction. This indicates much more openness and idiosyncrasy than previously thought. Hassenzahl [26] points out how products and services fulfill not only pragmatic (instrumental to a task) goals but also hedonic ones – those related to users’ self-expression and identification.

In addition to examining the factors of overall UX or goodness, researchers have analyzed the temporal structure of UX. Recently an international group of experts released a white paper outlining perspectives for UX research [55]. This paper provides a hierarchical definition of UX and emphasizes the importance of memories in understanding and defining experiences. In this view, momentary user experiences are “raw” emotional experiences, which become episodic user experiences through later appraisal of the raw experiences. Collectively all momentary experiences create a cumulative user experience. This defines UX as an “outcome and memories of an experience” [55, p. 7], as illustrated in Figure 1.

Figure 1. The structure of UX: anticipated, momentary, episodic, and cumulative UX. An adaptation of Fig. 2 in [55]

Affect
Emotions have only recently found their place in HCI [9, 49]. Emotions evoked by products and services are an important part of their use [e.g. 28, 48]. According to Höök [30], three distinct approaches to emotion among HCI and interaction design researchers can be distinguished. She calls them affective computing (e.g. as in [9, 44, 49]), interactional account (e.g. [2]), and the technology as experience view (e.g. [40]). This paper follows the first one by reviewing psychological theories and presenting a cognitivist model of affect and the experience of time. In particular, we deal with emotions, those short-lived feelings typically described categorically as anger or joy, or along the dimensions of arousal and valence [9].

Treatments of time in human-computer interaction
Time perspectives in HCI typically anchor to the concept of a task. When interacting with computers, the user is completing a primary task, which involves a series of interactions, a cycle of user inputs, and system feedback. System response time (SRT) refers to the time taken by the system to provide feedback for the previous user input. User response time (URT) is correspondingly the time that it takes for the user to provide new input after system feedback. SRT and URT can be measured objectively. SRT has been considered as an indicator of quality of service for a system [62].

Different types of tasks involve different expectations regarding SRT. The main division can be made between control and complex tasks [11]. Interaction can sometimes be complex, or conversational, like a dialogue between two humans [60, 61]. A conversational nature means that what we ask the other person determines how quickly we expect to get an answer. In spoken communication, short delays up to three seconds are normal [41]. But most interactions are “micointeractions,” simple control tasks in which we desire immediate feedback (short SRTs) to maintain a sense of control, for example when moving a pointer. This difference is crucial because complex tasks do not have a single correct SRT [11, 62], unlike control tasks [60, 62].

User adjustment
It is known that people actively regulate their emotions [22] through various means. Given the negative emotions associated with waiting, it seems natural that people find ways to avoid waiting (e.g., focus attention on another task). The few existing studies of multitasking [46] and task switching [66] imply that this user adjustment is normal behavior. For example, Flamm [14] interviewed public transport users who filled their travel time by socializing, relaxing, and being productive.

One can also consider how users “attune” to system behavior [53]. Tunable systems provide information about their behavior and behave predictably thereby enabling users to adjust to them. Conn [6] discusses similar behavior triggered by “time affordances.” Dabrowski and Munson
New concepts pertinent to time perception

We discuss two concepts to help understand the relationship between response times and episodic UX. The first is subjective response time, which closely resembles the concepts of subjective time progression [58] and introspective response time [8]. It is based on a user assessment of SRT. It is a perception of system responsiveness [60] that ranges from user unpreparedness (the system is too fast) through direct control to user in waiting (the system is too slow).

The second new concept we propose is subjectively experienced time (SXT). SXT is the assessment of system response timeliness and refers to how the subjective RT makes you feel. SXT is influenced by several factors, including past experiences and the present cognitive and affective state. SXTs are individual and context dependent (cf. [2]). For instance, if users have learned that delays occur in association with booking a ticket online for a popular concert, activating a virus scanner, or downloading a big file in the background, this will influence expectations. The subjective response time remains the same, but the SXT changes because the user can make sense of the delay and attribute it to an external source, influencing the subjective, emotional experience. Our claim is that for the needs of designing for UX, SXT is more important than SRT or URT [11], because SXT reflects the overall experience and extended waiting can provoke increasing levels of negative emotion, even anger.

HOW WE EXPERIENCE TIME

Humans have a capacity to maintain a daily rhythm and keep track of time. There is an accumulating psychological and neuroscientific body of evidence on how this happens. In this paper we are mostly interested in the functional characteristics of time perception. It is known that we can detect intervals between events both explicitly and by reporting duration estimates, and implicitly by completing tasks that require timing, for instance when playing an instrument. This performance is based on innate rhythms and external cues and seems to be a shared capacity for both explicit and implicit timing [50]. There are multiple complementary and contradictory theories of time perception [13, 42].

Psychophysical laws

Time estimation is considered to be subject to Weber’s law [19]. This law states that the relative difference between two compared quantities determines how easy they are to distinguish. The key concept is just noticeable difference (JND, also known as Weber’s fraction). A difference in two quantities can be noticed when the difference exceeds the JND level. Seow (2008) proposed that for HCI, the rule of thumb, JND, for time perception is 20%. However, Grondin [20, 21] has recently argued that Weber’s fraction for time perception is nonlinear and it is different for short durations, more likely in the range 10-15%. This is in line with studies [34] that found with a task duration of at least 1.5 sec. that the fraction was around 12%.

Vierordt’s law [38] is also relevant. It states that long time intervals are underestimated and short intervals are overestimated. The estimates are only accurate between 2 and 4 seconds. Note that this psychophysical research has mostly involved intervals of a maximum of ten seconds.

Psychological models: Scalar estimation theory

For this paper, we have chosen to utilize one widely recognized account, the scalar estimation theory [SET; 19, see 20], which explains how some animals, including humans, make absolute and relative time judgments. This model is supported by neuroscientific evidence from humans [e.g. 51] and has proven successful for modeling several empirical data sets. We propose a variation of the SET model (Fig. 2 next page). The variant integrates more recently introduced parts of time psychology with the SET model and, to helps to understand the experience of time.

The main changes in the model concern the inclusion of attention, arousal, and appraisal. In effect, we are adding an affective component to an informational account of experience [2]. This brings our model closer to the “technology as experience” view, by emphasizing continuous engagement and sense making [74].

SET proposes that time judgments are dependent on the functioning of an internal pacemaker, which produces pulses to be detected by an accumulator. The number of pulses collected by the accumulator counts towards the perception of elapsed time, enabling time estimation. This measurement mechanism relies on short-term memory but it is complemented by an assessment and decision-making process. In this process the new time estimate is compared to ones retrieved from long-term memory, the [42] reference expectations about how long this activity should take [12, 20]. This allows people to assess durations in both objective (seconds) and subjective (short, long) terms.

Emotional state, or prominently arousal, influences the rhythm of the pacemaker. Increasing arousal speeds up the pacemaker, causing more pulses to be sent within the same time. Arousal is modulated by emotional stimuli [45], direct stimulation of dopaminergic receptors [12], and changes in body temperature [72]. The modulating influence of affect on time perception in laboratory experiments seems short-lived. For example, overestimation bias disappears after two seconds [1]. However, highly arousing events can also be overestimated in retrospective estimation [64]. Arousal can also be temporarily modulated by diurnal variation [52]. This can be one reason why people are less tolerant
for delays during the lunchtime than later on in the day [60] and more accident-prone in the morning and during the afternoon “lows.” [31]

**Figure 2. A variation of the scalar estimation theory presented by Gibbon (1977) and reproduced by Droit-Volet & Gil (2009) for human time perception. Thin arrows represent information flow. Block arrows indicate modulation.**

Attentional effects are distinct from affective ones. The role of attention in the model can be clarified by thinking of time estimation as an independent task [8]. The allocation of attention to the timing task determines the awareness of time passing, i.e. time feels longer when you pay attention to it. Time estimation does not require our full attention, but the more occupied we are elsewhere, the less the pulses accumulate and the less we notice time passing [20], as if a connection between pacemaker and accumulator was switched off [71]. An empirical study supports this: “introspective estimates of time spent on a task tightly correlate with the period of availability of central processing resources.” [8, p. 1110] Hence some mental processes are hard to time subjectively because they take up all attentional resources. This explains the changed perception of time when people are highly engaged in a task [7, 10]. Together these components, pacemaker, accumulator, long- and short-term memory stores, allow relative and subjective time estimation, subjective response time in the HCI context.

The model of the experience of time is incomplete without a decision-making and actuation part that we call subjective assessment. Even the pigeons in Gibbon’s (1977) study “estimated” time by making a forced choice. However, in human SET variations, complex appraisal mechanisms are not commonly included, even though it is argued that all time-related judgments are affected by appraisal [12, 58]. Wright and McCarthy [73] point out how experiences with technology involve processes of anticipation, reflection, and interpretation, in the process of dialogical sense making based on past experiences. This also happens in the psychological appraisal theory of emotion which describes how people judge the pleasantness of experiences after the experience based on their interpretation of the situation [18, 54]. This theory assumes emotional reactions are an outcome of an evaluation process in which the initial visceral feelings and percepts are interpreted against all available information. Appraisal is necessary in understanding how we feel about the outcome of time estimation, thus we rely on the appraisal theory of emotion.

In the modified SET model, appraisal receives input from time estimation, past experiences, and information available from the environment. This process counts towards the feeling of subjectively experienced time (SXT), a meta-cognitive judgment [58]. Interpretation of the time estimate, rather than the subjective estimate alone, determines the emotional reaction and the same situation may evoke a different SXT depending on the appraisal (see 3.4.1 for an example of how information about time duration can modulate the appraisal of task enjoyment). In computing tasks, the number of possibly relevant factors is huge and much depends on how well users understand the operation of the system – how useful their mental models are.

This variant of SET is of limited utility for the retrospective assessment ("how long did it feel like?") associated with episodic UX. This also means that the timing information is not normally encoded because it is not in the locus of attention during task execution. According to Droit-Volet & Gil (2009, p. 1946), “the retrospective temporal judgment is then reconstructed on the basis of the nontemporal information retrieved from memory.” This connects time perception to emotions and makes it vulnerable for interventions which claim, for instance, that duration was shorter than experienced [58].

**GUIDELINES FOR SYSTEM DEVELOPMENT**

We provide six guidelines for designing for UX with consideration for SXT. The guidelines are presented in Table 1 and are divided into design and evaluation processes, and user adjustment support guidelines. Guidelines include implementation examples, but are otherwise intentionally abstract so that they could be implemented in many types of systems, foremost in everyday computing.

<table>
<thead>
<tr>
<th>Design and evaluation process</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Adhere to tolerance thresholds</td>
</tr>
<tr>
<td>2. Optimize the system for reduced SXT</td>
</tr>
<tr>
<td>a) Changes in SRT must exceed the JND</td>
</tr>
<tr>
<td>b) Prioritize short wait times</td>
</tr>
<tr>
<td>c) Reduce the number of waiting periods</td>
</tr>
<tr>
<td>d) Prioritize the loading order of display elements</td>
</tr>
<tr>
<td>3. Turn waiting time into occupied time</td>
</tr>
<tr>
<td>a) Provide alternate tasks</td>
</tr>
<tr>
<td>b) Provide passive entertainment</td>
</tr>
<tr>
<td>4. Include SXT in design requirements and evaluation criteria</td>
</tr>
</tbody>
</table>

**User adjustment support**

| 5. Inform the user about the wait quickly and accurately |
|   a) Clarify and justify waiting |
|   b) Provide actionable waiting estimates |
| 6. Inform the user about system readiness |
|   a) Give immediate feedback when waiting is over |
|   b) Indicate system readiness accurately |

**Table 1: Six guidelines for improving the experience of time**
1. Adhere to tolerance thresholds

The central concept about time in user requirements engineering is the tolerance threshold. Threshold refers to the typical time (SRT) one can wait before perceiving a delay and experiencing waiting time. Threshold expectations are derived from identical or similar past experiences. Designers should become aware of the tolerance thresholds that users have for the current system and strive to meet or better them [60, 62].

Several concrete proposals exist about tolerance thresholds as they have been investigated since the late 60s [43]. Many of the modern guidelines define absolute time limits for interaction [63]. Shneiderman and Plaisant [62] and Seow [60] follow this idea, but label categorized interactions and thresholds differently (see Table 2). They elaborate the complex vs. control task division presented earlier (but see [11]). Seow’s [60, p. 52] categorization includes four different types of tasks: instantaneous, immediate, continuous, and captive. Each category has characteristic thresholds. Shneiderman [62] relies on a similar rationale, but with different labels shown below.

<table>
<thead>
<tr>
<th>Interaction type</th>
<th>Threshold(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instantaneous</td>
<td>100 - 200 ms</td>
</tr>
<tr>
<td>Immediate</td>
<td>0.5 - 1.0 s</td>
</tr>
<tr>
<td>Continuous</td>
<td>2 - 5 s</td>
</tr>
<tr>
<td>Captive</td>
<td>7 - 10 s</td>
</tr>
</tbody>
</table>

**Table 2.** Upper and lower thresholds for acceptable SRTs in different types of interactions recommended by Seow [60] and Shneiderman [62].

Threshold examples can be helpful rules of thumb for design. However, the thresholds and SRTs are insufficient to predict SXT, because there is still arousal, attention, and the environmental information to consider. Another challenge is the correct classification of the interaction type. The designer needs to understand which type of interaction they are dealing with from the user perspective.

Hence the first step for the UX specialist is to identify possible wait times and related interaction types. How long are SRTs in objective terms? Do system responses occur within the threshold times? What are the resulting SXTs? A system walk-through is necessary to identify the steps during which SRTs might exceed the tolerance thresholds. This should be based upon realistic use cases. After charting the SRTs, one should identify the causes and consequences of waiting for later optimization. Waiting at different steps of the task has different overall impact. At worst, initial unresponsiveness may cause the task to be postponed or abandoned [6], supporting the heuristic that it is important to get started [39] and done quickly [17, 24, 25] – or at least appear to do so.

2. Optimize the system for reduced SXT

If the system leaves the user waiting, the system design should optimize the SRTs to curb the negative effects.

A. Changes must exceed the JND

The time perception can be manipulated by reducing objective waiting times. Weber’s law can be utilized while deciding when an upgrade justifies the cost. JND sets a limit for the minimum modification that the user might perceive. However, this does not mean that a change equal to a JND would suffice; the desired threshold for the interaction type should be meet (see previous section). We suggest using 1/8 (12.5%) as the rule of thumb for JND for durations of two seconds or more and 1/5 (20%) for shorter durations. This differs from Seow’s [60] recommendation but seems more compatible with psychological data [21, 34]. We have calculated a set of JNDs based on a nonlinear Weber fraction for Table 3. For instance, if a web page previously took 10 seconds to load, the JND is 1.25 seconds and the loading time should be reduced to at least 8.7 seconds to make any difference for SXT.

<table>
<thead>
<tr>
<th>Duration</th>
<th>Fraction</th>
<th>JND</th>
<th>New duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 ms</td>
<td>1/5</td>
<td>20 ms</td>
<td>80 ms</td>
</tr>
<tr>
<td>1 s</td>
<td>1/5</td>
<td>125 ms</td>
<td>875 ms</td>
</tr>
<tr>
<td>3 s</td>
<td>1/8</td>
<td>375 ms</td>
<td>2.6 s</td>
</tr>
<tr>
<td>5 s</td>
<td>1/8</td>
<td>625 ms</td>
<td>4.3 s</td>
</tr>
<tr>
<td>10 s</td>
<td>1/8</td>
<td>1.25 s</td>
<td>8.7 s</td>
</tr>
<tr>
<td>15 min</td>
<td>1/8</td>
<td>1 min</td>
<td>13 min</td>
</tr>
<tr>
<td>30 min</td>
<td>1/8</td>
<td>3 min</td>
<td>26 min</td>
</tr>
<tr>
<td>60 min</td>
<td>1/8</td>
<td>7 min</td>
<td>52 min</td>
</tr>
</tbody>
</table>

**Table 3.** Just noticeable differences for some SRT durations.

B. Reduce the number of waits

Prefer having fewer but longer waits. Based on Vierordt’s law, it seems better to have a one-minute wait than six 10-second waits. If the users switch tasks or multitask during the wait, this will also reduce URT as more time will be cumulatively spend on orienting to alternative tasks. Sometimes it can be possible to reduce waiting time by considering the arrangement of interactions. This could be done by parallel execution of slow, system-intensive tasks in the background with interactive, less intensive tasks in the foreground. This could help to reduce the number of waits. For instance, installation of an operating system involves repeated dialogues with the user (names, passwords, and such), with considerable waiting time between them. Running dialogues and installation in parallel might reduce the waiting times. This has already
happened: for example, proofreading used to be a separate word processor task, but now occurs while typing [11].

C. Solve the root cause
This is a common-sense principle for optimizing systems. To maximize the return (positive return on investment), one should look for wait times that have a common root cause. This will create a cumulative benefit.

D. Prioritize short wait times
Guideline 2.a implies a logical way to optimize systems for SXT. Taking into consideration Vierordt’s law about how short durations are overestimated and long durations are underestimated [38], it is more effective to optimize initially short waiting times than the longer ones. If absolute reductions of time require a comparable effort on different time scales, then it is also better to optimize short waits in which the JND is easier to exceed.

E. Prioritize the loading order of display elements
In dynamic loading schemes, the presentation order of interface elements can create periods of waiting. Web services exemplify systems in which both interactive and informative elements of the interface are loaded and displayed to the user with a variable delay. The trend is that the web browser displays elements as soon as they have been retrieved from the Internet. The appearance order should be prioritized so users can utilize the page while other elements are still being loaded. As of 2013, this feature has been exploited by many ad-driven websites.

3. Include SXT in design and evaluation criteria
SXT requirements should be included as a part of user requirements and user evaluation [cf. 6]. Identifying issues related to the experience of time could be combined with usability testing. Defining requirements for the existence of waiting times and their thresholds is straightforward. Participants can be asked to provide time estimates, or just asked about noticing delays or waiting. Retrospective probing after the task seems acceptable to recall episodic UXs. We have collected a list of questions for UX professionals which can be used as heuristics to Table 4.

For UX specialists, we recommended starting from the relevant use cases of the system. This can be used to identify the points in an interaction sequence at which the user has to wait for feedback or action (Guideline 1). When the waiting time exceeds three seconds, we recommend providing feedback about the situation (Guidelines 5 & 6).

4. Turn waiting time into occupied time
A: Provide alternate tasks
If waiting does occur, activities that divert attention from waiting can be made available. In some cases, it may be possible to provide the user with a secondary task that fills up the waiting time, taking attention away from the “waiting” task, as illustrated by the innovation legend of installing mirrors in elevators to “make them go faster.” At best, this task might be something relevant to the main task [39] or other pastime activity, such as gaming [69].

B. Provide passive entertainment
A common practice with traditional services is to equip the waiting area with a TV screen, radio, or dedicated background music. Computer games commonly include game-related visuals as loading time entertainment. Music is widely used and easy to utilize. Its positive impact is derived from both attentional and affective influences. Background music has been utilized to modulate main task performance, although not so commonly in computing tasks [see also 16, 47]. The influence of background music is idiosyncratic. The objective characteristics of music do predict receptivity, but the emotional impact of music is highly dependent on individual preferences and familiarity [59]. In Western culture, instrumental music is the universally utilized background score [23], although it is known that commercial format radios [70] play carefully selected music to appeal to sharply defined consumer segments because of the considerable differences in receptivity across audiences.

As a design recommendation, leaving a chance for the user to choose the pastime activity is probably the best diversion. Assuming we can predict the delay, the system could provide a few alternatives, say videos of similar length from categories related to the main task, for a user waiting for a response from an Internet service.

Table 4. Questions for heuristic evaluation and user evaluation

<table>
<thead>
<tr>
<th>For heuristic evaluation</th>
<th>Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>I Are waiting times over 3 seconds likely to occur?</td>
<td>1</td>
</tr>
<tr>
<td>II Are the waiting times known? If so:</td>
<td></td>
</tr>
<tr>
<td>Is a waiting time estimate displayed?</td>
<td>5</td>
</tr>
<tr>
<td>Is the waiting time justified?</td>
<td>5</td>
</tr>
<tr>
<td>What is the distribution of delays like?</td>
<td>5</td>
</tr>
<tr>
<td>III Can the interactive session expire? If so:</td>
<td></td>
</tr>
<tr>
<td>How does the design support multitasking?</td>
<td>5, 6</td>
</tr>
<tr>
<td>How does the user know the time limits?</td>
<td>5</td>
</tr>
<tr>
<td>IV What happens if the task is switched or interrupted while waiting?</td>
<td>4, 6</td>
</tr>
<tr>
<td>V How does the user know that the waiting has ended?</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For user evulation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VI Can user threshold expectations be confirmed with users?</td>
<td>5</td>
</tr>
<tr>
<td>VII How will users react to the alternate task or pastime?</td>
<td>5</td>
</tr>
</tbody>
</table>
The next set of guidelines deals with **user adjustment**. Designers need to consider design for task switching and multitasking now possible on all computing platforms. Systems that have waiting times should provide adequate information about the wait and the consequences of a delayed return to the task (see Guidelines 5 and 6). This provides the user with a chance to cancel the task and prevents accidental interruptions.

Implementing the traditional usability property of system **transparency** [57] and revealing all time-relevant information helps to maintain awareness of the progress of the main task (see 4.8) while the user is attending to a secondary task. As an example, consider delays in an online manuscript submission system. Creating a PDF proof file might take approximately 2 minutes before it needs to be examined and approved by the user, but after 10 minutes of user inactivity, the system will close the session and abort the initiated transaction. If users are informed about these limits, then they have the opportunity to execute secondary tasks, get a coffee, read news, etc., in the 2-10 minute time window without waiting and trusting that the interaction can still be completed as planned. The questions UX specialists should ask are: what happens if users switch to doing something else while waiting? How long can the system keep on waiting, what are the consequences of the user returning late? Can we warn the user about the possible time-outs and discourage multitasking in critical use cases?

**5. Inform the user about the wait:**
**Quick, accurate, and justified messages**

Always justify and clarify waiting. If the SRT exceeds the user’s threshold, the user should be promptly informed about the wait. Clarification of the situation combats user frustration and useless attempts to continue interaction when it is not yet possible. This message should indicate the reason why the user is put on hold and for how long [cf. 39]. The justification does not make the waiting go away, but can give it a different meaning and change the appraisal of the situation.

The user should be given actionable **waiting time estimates** and **indications of progress**. Waiting dialogues are used to serve this purpose. Information about the wait, preferably about the time remaining [58], supports user adjustment. If progress indicators are used, their design should be considered from the user’s perspective [24]. We recommend displaying waiting times that are realistic, empirically measured, or accurately modeled. The estimate should be positively biased, that is, more than the expected (average or mode) waiting time. The rationale is that it is better to underpromise and overdeliver. Ideally, the estimate should be determined by a loss function (consequences of mismatch in observed vs. expected wait) and matching it to the observed distribution of waiting times to find an optimum. An alternative to exact estimates is **anchor times** [67], which people naturally use in describing and estimating intervals. Seow [60] proposed a time anchor matrix with anchors 1, 2, 3, 5, 10, and 15 seconds and 10, 20, and 30 minutes, and recommends providing estimates as ranges between these anchor points or upper limits.

Some currently utilized ways to inform the user about the wait are illustrated in Figure 3. The screenshot shows that Priceline.com online travel agency informs the user promptly and humorously but not necessarily adequately about the waiting. The screen provides a dummy **system activity indicator**, a line of moving dots, but no information about the expected duration of the delay or what to do if the user wants to cancel the transaction. The dialogue window of a Windows file utility xplorer2 gives a more actionable example with a real progress indicator and an anchored remaining waiting time estimate (“about 25 seconds”), and informs the user about what the system is doing.

**6. Inform user about system readiness accurately**

Give prompt feedback to indicate that the wait is over. If a separate waiting dialogue (Guideline 5) or equivalent acknowledgment is not provided, the interpretation of the delay is solely the user’s responsibility. Without an appropriate indicator of system responsiveness, users may feel that they are still waiting when in fact the system is waiting for them. Provide the **end-of-wait feedback** only when the system is really ready for input. Favor a rapid conclusion as users tolerate slow system responsiveness worse towards the end of a wait [24]. For instance, old Windows versions displayed the desktop at a relatively early phase of starting the operating system deceptively.
before the user was able to use it. In contrast, OS X displays the desktop after sleep when it is ready for use.

**DISCUSSION**

We enjoy experiences more when they seem to go by faster [5, 58]. Interruptions while interacting with technology can effectively hamper our experiences. The goal of this paper has been to explore how interaction design can support smooth time-related experiences and consequently improve UX. We argue that subjectively experienced time outperforms system response time as an explanatory concept when trying to improve a computing system in terms of UX. This paper has focused on how to influence SXT and curb damage from unavoidable pauses.

We emphasized that SXT mediates overall UX. Therefore designers need to understand why people perceive SXT as they do. With the help of guidelines such as presented here, we can devise interventions to influence UX and support users in having positive, empowering experiences with technology (cf. [2]). We believe that the concept of SXT opens up a useful way to understand and quantify the quality of service [62]. For design purposes, SXT reminds us that beyond optimizing systems and reducing SRTs, there are alternatives to improve SXT through design. This can mean rearrangement of the workflow, supporting user adjustment with accurate information, or providing activities to bridge the waiting time. For instance, Houston airport successfully reduced customer complaints for slow baggage claim by filling waiting time with activity: they increased the gate distance so the time previously used on waiting was now spent on walking [65].

The work we presented has limitations. This work could be criticized for falling squarely in the traditional cognitivist, affective computing framework. Indeed, we have built our model and design implications on the literature on individual psychology and cognition. However, we have expanded the model modestly beyond the individual to assimilate some features of alternative approaches [2, 40]. Nevertheless, this account does not accommodate the influences of social interactions, non-task-relevant environments, or less pragmatic tasks [26] very well. However, we do not see our philosophy as being in direct conflict with “slow computing” – it’s a matter of determining what the ultimate, overall goals of the system are. Simple and feedback tasks (Table 2; [60]) within a “slow” system might still be designed along our guidelines.

There are also disclaimers to details of our work. Many of our guidelines assume that the system response times are relatively stable, without a considerable variation between the use sessions. User adaptations with systems can only develop if the system behaves consistently [53]: e.g., if turning on the computer always takes exactly five minutes, this time can be easily appropriated for a secondary task. If it has unpredictable delays, it will be more difficult to cope with. Our review has raised several issues that could be, but have not been, examined by empirical studies. Consequently some of our inferences would benefit from further research.

**Future work**

The concept of SXT should be tested and developed. We should explore appropriate ways to measure SXT. So far we have only discussed subjective response times (measured from too fast to too slow) and not provided any specifics about recording the emotional responses relevant to SXT. For the latter, different types of emotion instruments utilized by psychologists, e.g. the self-assessment manikin [68], could be used together with probes focusing on the cognitive appraisal of the situation. There is also the question of prospective vs. retrospective evaluation, i.e. when should SXT be assessed? Considering the existence of the peak-end bias [17], retrospective evaluation seems more important and should bring forth especially negative episodic UXs.

User adjustment is maybe the biggest empirical issue relevant to time factors and overall UX. The current understanding in HCI of user adjustment and adaptation to system responses is outdated. The extensive review of Dabrowski and Munson [11] did not find any references relevant to adjustments published after 1994. The world of personal computing has since changed remarkably, from command lines to GUIs, from desktops to mobiles. Users are continuously facing the temptations and challenges of multitasking, task switching, and interruptions [66] which are not always accounted for in design.

**CONCLUSION**

We have presented several interventions for designers to curb boredom should system responses become slow. Our guidelines build upon the concept of subjectively experienced time and suggesting ways to support users in experiencing technology. These aim to improve the overall UX. We have also motivated a lot of additional research.

**ACKNOWLEDGMENTS**

This work was supported by the research projects Lutus (Technology Industries of Finland Centennial Foundation) and Theseus II by the Finnish technology agency Tekes.

**REFERENCES**


