Interacting with Computers 23 (2011) 555–564

Contents lists available at ScienceDirect

Interacting with Computers

journal homepage: www.elsevier.com/locate/intcom

40 Years of searching for the *best* computer system response time

Jim Dabrowski [a], Ethan V. Munson [b,*]

[a] University of Wisconsin-Rock County, 2909 Kellogg Avenue, Janesville, WI 53545, USA
[b] University of Wisconsin-Milwaukee, Computer Science, PO Box 784, Milwaukee, WI 53201, USA

ARTICLE INFO

Article history:
Received 13 May 2010
Received in revised form 29 May 2011
Accepted 31 May 2011
Available online 12 June 2011

Keywords:
Human computer interaction
System response time
Software response latency

ABSTRACT

For over 40 years, system response time has been a topic of interest and controversy in computer science. Since the late 1960s, the field has seen numerous studies conducted and articles written addressing the issue. Many factors were measured in these studies including: users' accuracy and error rates with different levels of system response time, user performance speed and the efficiency of the commands used, how user interactions with the computer changed as a result of changes in system response time, how their bodies reacted physiologically to those same changes and even how happy, satisfied, anxious or annoyed they were as system response times changed.

In this paper, we summarize the major issues in system response time research and look at what can be concluded from them. Generally, researchers have suggested specific response-time guidelines based on the complexity of the task or the type of interaction with the computer. We suggest that rather than system response time being task- or expectation-focused, instead interactions with a computer fall into two categories: control tasks and conversational tasks. For control tasks, immediate response times are necessary for optimal user performance whereas for conversational tasks, some delays may be necessary to maintain the optimal pacing of the on-going conversation. The location and duration of these delays will depend on **both** task complexity and user expectations. Future system response time research is needed to further quantify limits of delay detection, and the location and duration of inter-task delays to optimize user performance and satisfaction with computers.

1. Introduction

Ever since people started using computers they have wanted to make them faster. As a result, CPU speed and overall system speed has increased in a never ending quest for a computer that is "fast enough." In some regards, computers will never be fast enough—with every increase in processing power and speed come improvements to applications that demand ever greater resources from the computer. In other respects, they are already more than fast enough. Computers today can easily perform most simple tasks faster than users can think about and respond to them. The popularity of so called "netbooks" shows that personal computers may have, for the first time in their history, reached a point where they are finally fast enough for the average user. These smaller computers use CPUs with clock speeds significantly slower than the top of the line chips, hard drives with slower access times, and integrated graphics chips with poorer performance than would be provided by a dedicated graphics card. Despite all these performance drawbacks, netbook sales continue to grow relative to sales of desktop PCs and more traditional laptops. Consumers have clearly made the decision that for some tasks, computers today are more than fast enough.

The search for the best computer system response time effectively began in 1968 when Robert Miller wrote an article entitled, "Response time in man–computer conversational transactions" (Miller, 1968). In that article he suggested that human–computer interactions were akin to conversations between two people. The user issues a command to the computer and waits for a response. Based on that response, the person will issue further commands until the "conversation" is over. He suggested several limits of system response time (SRT) for various types of interactions with a computer based on expectations people have while in conversations, short-term memory limits and how delays within a conversation can stress those limits, and the recovery time needed from the closure of one task until the start of another. He described 17 types of human–computer interactions and provided guidelines on how long a delay should be acceptable to a user. Although not based on empirical investigation, Miller's taxonomy provided a good starting point for a wealth of investigation into just how fast computer systems should respond to user actions.

At that time, the major focus of system response time research was on how quickly main-frame computers needed to respond to user requests for interaction over the local network. Whether a

* Corresponding author. Tel.: +1 414 229 4438; fax: +1 414 229 2769.
E-mail addresses: james.dabrowski@uwc.edu (J. Dabrowski), munson@uwm.edu (E.V. Munson).

0953-5438/$ - see front matter © 2011 British Informatics Society Limited. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.intcom.2011.05.008

computer needed to respond to a request within 2-s was a hot topic of debate (Kosmatka, 1984; Lambert, 1984). In the 1980s, computers and their data moved from the back room onto the desktop and the debate dwindled. Since that time, the growth of the World Wide Web and the use of Internet-based applications has caused a renewed interest in computer system response times.

Today the line between a desktop application and a network-based one is blurring. In some cases such as Google Docs, Facebook and web-based email, the applications are contained entirely within the web-browser. With the use of the latest HTML and CSS, developers can create web-pages that completely mimic a standard desktop application. In other cases the applications are traditional programs, but they interact with and present data from the Internet almost without the user's awareness. Nowhere is this more apparent than in today's "smart phones" where people can have applications that continually retrieve data from "the cloud" and present that data to the user for manipulation. Changes to the local data can be pushed back to the Internet without any awareness on the part of the user. It is precisely because of this return to network-based computing that system response time continues to be an important issue in computer science research.

Today's software engineers need to be aware of a network's Quality of Service (QoS) when designing their Internet-based applications. QoS generally refers to a network's ability to guarantee a certain level of performance in the delivery of data. If networks cannot deliver data quickly enough, applications may seem unresponsive or the data they deliver may suffer from jitter or other degradations of quality. But from an end user's point of view, QoS and system response time are essentially the same thing, even though a technical professional might consider QoS to be a network issue and system response time to be a computer issue. An end user is only concerned about whether delays originate on the network or on the computer when trying to decide what to fix or replace.

So for software engineers, the fundamental questions remain the same: How quickly does a software system need to respond to user interaction? If the software does not respond quickly enough, how will this affect user performance and satisfaction with the software? The advent of Internet-based applications has simply introduced a new factor into the system. Oftentimes, software developers have no control over network performance or network-related delays. Some end users may be on slow, unreliable networks while others may be on fast, high-bandwidth networks. As a result, it is even more important for today's software engineers to be aware of the effects of system response time on user performance and satisfaction with computer systems. For example, when network delays are unavoidable, how can application interaction be altered to hide or otherwise mitigate their impact? Should interactions with the local application be as fast as possible at all times or should they be altered to match network speed in order to avoid variability in the responsiveness of the application? It is clear that computer system response time continues to be a relevant issue more than 40 years after researchers began studying it.

The remainder of this paper is organized into the following sections:

- *Review of research:* Rather than being organized as a chronology, the relevant research is presented according to how human performance during interaction with computers was measured in each of the studies reviewed.
- *Existing SRT guidelines:* This section reviews and critiques the two existing models of computer system response time and their suggested response time limits.
- *Proposed categorization of human–computer interactions:* This section proposes a new model of computer system response time suggesting that all interaction with a computer can be divided into two categories: control tasks and conversational tasks. This new model attempts to fuse and improve upon the existing models.
- *Final words:* A short summary of the paper.

2. Review of research

Many empirical investigations have been carried out over the past 40 years attempting to determine the best system response time (SRT) and what factors contributed toward that best. As might be expected, no single definition of best was used across all studies—each study had its own ways of measuring human performance with computer systems. Some studies were naturalistic observations—researchers would observe peoples' performance on computer systems in the workplace or other environment and correlate that performance with observations and sometimes manipulations of SRT. What data were collected was often constrained by the capabilities of the computer system in question. There were also laboratory experiments in which subjects were asked to perform specific tasks with a computer while the experimenter manipulated aspects of the computer system.

Because of the diversity of approaches to SRT research, a simple chronological review would lack coherence. It is more profitable to examine the research based on how human performance during interaction with computers was measured because the results show greater consistency when viewed in this manner. Several distinct categories of human performance measures have been used in SRT research and within these categories, different questions have been asked and answered. Individual studies often used measures from multiple categories. These categories and questions are as follows:

Errors: When interacting with computers, people are bound to make mistakes—enter the wrong data, issue a bad command, click on the wrong interface element. Does SRT have any effect on the commission of these errors or are error rates independent of SRT? Is there a lower-limit of SRT below which error rate is unaffected or is there an optimal SRT for different tasks that will result in the fewest errors?

Productivity: Of paramount interest in many environments is how quickly can users finish a given task with a computer or how many tasks they can complete in a given amount of time. It seems intuitive that the less time users spend waiting for the computer system to respond, the faster they will get their job done. But is this always true? Will users always respond as soon as the computer is ready or does some amount of delay actually fit more naturally with a user's pace of interaction such that they will accomplish their task more quickly?

User adjustment: Computer users have shown over the years that they can adapt to just about any computing environment. Of interest here is what is the interaction like? How do changes in computer SRT change the manner in which users interact with their computers if at all? Do faster computer response times encourage one type of interaction while slow computer systems force another?

Psychological effects: When interacting with a computer system, everyone at some point becomes angry, frustrated and annoyed with how slow it is responding. At what point do delays in SRT turn happy, satisfied users into angry, frustrated, dissatisfied users? Will users be more tolerant of delays while engaged in one type of task while less tolerant of delays when engaged in another? Finally, can users be mollified and lulled into tolerat-

ing delays by providing them with feedback when waiting for the system to respond?

Physiological effects: Do delays in SRT cause users stress and anxiety? Can this stress and anxiety be measured physiologically? Some studies have used heart-rate, respiration, electrical activity of the brain, and galvanic skin conductance as measures of the amount of stress a user is experiencing while using a computer system in an attempt to determine if there exists an optimal SRT for such interactions.

The findings in each of these areas are discussed in detail below.

2.1. Errors

When measuring the effects of SRT on human performance with a computer, one of the most straight-forward measures is how many errors the person commits when engaged in a task. When positioning a cursor with the mouse or keyboard on a user interface item or within a document, how does SRT interfere with that positioning? When users are performing simple data entry, is there a response time that will produce the fewest submission errors? When the human–computer interaction is more conversational, do delays matter as much or does some amount of delay actually improve performance? Many studies have tried to answer these questions.

Delays in SRT have the greatest impact on users when those delays interfere with perceptual feedback and knowledge of the results of user input. If the computer system does not keep up with a user's simple actions such as positioning a cursor or typing, there is a clear relationship between the amount of delay and the number of errors committed. MacKenzie and Ware (1993) had subjects position a mouse cursor onto a target square located on the far edge of the screen while they delayed the redrawing of the cursor as the user was trying to position it. The delays inserted ranged from 8.3 to 225 ms. They found that even at the lowest levels of delay, there was a significant impact on user error rates and as the delay increased, so did the error rate. Similarly, Williges and Williges (1982) had subjects perform simple database entry and editing tasks. In part of their study, they controlled the rate at which the characters appeared on the screen after the user typed them on the keyboard (echo rate). Like MacKenzie and Ware, short delays of about 200 ms began to have an impact on the typing errors that users committed while editing and the users' error rates increased as the delays increased.

While delays at such an atomic level (key-presses, mouse-clicks, cursor movements) greatly interfere with human performance, when the delays occur during the flow of exchange between human and computer, the impact is less clear-cut. A common interactional task studied by many computer-scientists is data-entry. To this day, people spend a great deal of time typing data into a computer. A common method of studying this task is to require the subject to type in some sequence of characters or numbers and then press the enter or tab key to move onto the next one. The computer will respond that it is ready after some experimenter-controlled amount of delay. Several studies (Butler, 1983; Dannenbring, 1984; Martin and Corl, 1986; Teal and Rudnicky, 1992; O'Donnell and Draper, 1996) have examined this and the results have been pretty clear: delays do not affect the accuracy rate of people entering the data into the computer. In all these studies, subjects were required to type in data ranging from three to five characters in length with delays ranging from 0 to over 30 s. Many of the studies also varied the amount of delay between lines of data to determine what influence that would have on error rates. In all cases, there was no impact of SRT on the errors people made. Even when the data entry was but the first step in a two-part task of looking up information (Butler, 1983; McCain, 1993) delays inserted between the user pressing the enter key and the secondary results appearing on screen do not affect the user's typing accuracy or their accuracy in finding the information requested. It is as if the tasks were simple enough that subjects were able to maintain the mental continuity of the interaction or else the tasks were discrete enough for the users so that no continuity needed to be maintained.

When the task gets more complicated, or the flow of information between user and computer become more inter-dependent, the results begin to get more interesting. Barber et al. (1983) monitored the live work performance of telephone system engineers while they processed orders for configuring new service or modifying existing service. For any given task, the user had to issue a dozen or more commands to gather the information necessary to fill the order and then had to issue more commands to actually implement the change. The experimenters were able to adjust the SRT of the computer making it worse for several days (an average of a 14-s response time) and then returning it to its previous responsiveness (an average of a 6-s response time). They found that the number of errors made in processing an order by the operators was minimized when the SRT was in the 12–14 s range and increased at both higher and lower response times.

Several controlled experiments were also conducted in an attempt to determine if an optimal SRT existed for specific tasks. Kohlisch and Kuhmann (1997) and Schaefer (1990) had subjects perform a Sterzlinger task under different delay conditions. In a Sterzlinger task, subjects are required to scan a line of characters on a computer screen for a specific target (a gap appearing between identical letters), position the cursor in the gap using arrow keys and press another key to indicate they found the target. In this task, delay was inserted between the completion of one task the resumption of the experiment with the next, identical task. After several experiments Kohlisch determined that about 1.5 s was the optimal duration of inter-task delay for this task (Kohlisch and Kuhmann, 1997). Similarly, Thum et al. (1995) had subjects scan a grid of two-digit numbers for the presence of two targets. Subjects had to indicate whether none, one or two targets were present in each grid. Here too, they found a specific optimal SRT that resulted in the fewest errors committed on the following task by the subject. SRTs shorter or longer than this optimal resulted in significantly more errors.

Finally, the interactions between human and computer can become far more conversational and relaxed where delays in the responsiveness of the computer system can be seen as a natural part of the interaction or perhaps they do not interfere with the user's ability to move on with his or her work. In such cases, it again appears that delays in SRT have little to no impact on the number of errors that the users commit. In an experiment by Dannenbring (1984) subjects interacted with a computer in a psychotherapeutic manner. The subjects in this study typed questions or comments to the computer and the computer responded with a vague, reflective responses emulating a type of psychotherapy. In such a conversation with a person, one would expect there to be pauses of thought while one person is considering what was said and how to respond. It should come as no surprise that even lengthy delays had no influence on the number of typing errors the users committed while engaged in this task. In an experiment of a different nature, Grossberg et al. (1976) had subjects interact with a computer system to solve complex mathematical problems. To solve the problems in this experiment, subjects had to input several commands and periodically have intermediate results output to a printer or to a computer screen. After issuing such a print command, the user was able continue issuing other commands to continue solving their given problem. Delays of over a minute inserted between issuing a print command and the results appearing on screen or

on paper had no influence on the number of errors subjects made while trying to solve their problems.

When it comes to error rates during the flow of human–computer interaction, one finds the greatest effects at the extreme ends of the interaction. When the delays in the responsiveness of the computer system interfere with the simple appreciation of feedback and knowledge of results of user input, the effects are immediate and obvious. People cannot perform well when the computer cannot keep pace with their movements. Even minimal delays will have a significant impact on their performance. At the other end of the process are those types of interactions where the "conversation" is more slow-paced and delays are a normal, expected part of the interaction or when the task is such that any delays can be safely ignored while the user goes about his or her work. In these cases, even substantial delays will not have any effect on the number of errors the user commits. It is in the middle-ground where questions still remain. Clearly, for different tasks there can be an optimal delay that will minimize the number of errors a user commits. This optimal delay varies greatly from task to task and it remains a goal of researchers to identify and categorize those components of a human–computer interaction that most significantly contribute toward this optimal delay.

2.2. Productivity

Another area of great interest to researchers has been the effect of SRT on user productivity. Does making the computer system respond faster to user actions help users get more work done? If SRT is slowed, do users work disproportionally slower or can it speed up their performance? As with errors, there are different ways to measure productivity. One way is to measure the amount of work completed or the average amount completed in a standardized time period. Another is to simply measure the amount of time it takes a user to complete a given task.

When it comes to measuring amount of work completed, the results of the research are surprisingly clear: longer SRTs result in lowered productivity. A few studies measured the productivity of employees in the workplace. Barber et al. (1983) studied employees using mainframe computer systems to process telephone orders, while Kosmatka (1984) and Lambert (1984) studied computer programmers. In all cases, they found that as SRT increased, the number of transactions or commands issued to the computer decreased linearly as a function of SRT. Barber et al. even measured "productive transactions" (those that did not contain an error and that furthered the processing of an order) and found that productive transactions stayed relatively flat until SRT reached about 8 s and then it decreased steadily from that point forward. Lambert, in addition to measuring commands issued, also measured the number of function points and lines of code completed by the programmers. They found that those programmers with faster SRT completed more lines of code and more function points than those in the slow SRT condition. Kosmatka, however, challenged the belief that faster SRT results in more productive employees. He pointed out that the programmers in his study were not necessarily more productive even though they were issuing more commands. He felt that the commands issued were more trivial and that the employees simply employed different strategies as response time changed.

Several other studies measured productivity in the laboratory and came away with essentially the same conclusion whether the task was interacting with the computer in a conversational manner (Dannenbring, 1984), issuing a series of commands to solve a problem (Grossberg et al., 1976; Martin and Corl, 1986) or editing a document (Guynes, 1988). As SRT increases, users will issue fewer commands and may complete fewer tasks. The study by Martin and Corl (1986) is especially interesting since they also manipulated the difficulty of the task. In part of their study, subjects performed a simple data-entry task and they found that for these tasks as SRT decreased subjects were able to complete more tasks per hour. In another part of the study subjects were engaged in more complex problem-solving and for these tasks, changes in SRT had almost no effect on the same measures of productivity.

The results of studies investigating how SRT influences task completion time are not nearly as clear-cut. As with error rates, delays seem to have the greatest influence on completion time when those delays interfere with the perceptual process. In the MacKenzie and Ware (1993) study short delays ranging from 8.3 up to 250 ms significantly increased the amount of time it took subjects to move the mouse cursor onto the target. Furthermore, as the tasks became more difficult, the influence of those delays became even more pronounced. Similarly, Williges and Williges (1982) found that delays in the appearance of characters on screen after being typed coupled with delays in the cursor moving from one field to the next in a database data entry task caused significant disruption in the user's interactions and caused them to spend more time waiting and watching the system to make sure it was ready for them.

A mixed effect is seen when the delays do not interfere with the perceptual process, but instead occur between the repeated performance of identical tasks such as occurs in data entry. Dannenbring (1984) found that for a simple four-digit data entry task when SRT was subtracted out, total task completion time actually decreased as SRT increased. When searching for targets in lines of letters (Schaefer, 1990; Kohlisch and Kuhmann, 1997) or 2-D grids of numbers (Thum et al., 1995), there may be an optimal response time that produces the fastest performance on that next task, but the results are not completely clear. Kohlisch and Kuhmann (1997) and Thum et al. (1995) both found that performance was worst at the lowest levels of delay and improved as delay increased up to a point. Beyond that point, performance again worsened, but the worsening was not significant. Schaefer (1990), on the other hand, found a linear decrease in performance as SRT increased but the delay range was studied was narrow and may not have been large enough to demonstrate any possible worsening of performance as delays increased.

If the computer system is simply a little slow in responding to a user's initial action, but responds normally after that, users appear to be able to tolerate this and not allow it to have any significant impact on the amount of time they take to complete their given task. For example, Goodman and Spence (1978, 1981, 1982) found that delays ranging from .16 to 1.8 s in the responsiveness of an on-screen control in a graphical user interface had no effect on the total amount of time subjects took to complete their given task. Similarly, 30–100 ms delays in the responsiveness of a cursor to the initial press of an arrow key to position it had no influence on total time it took to complete editing a document (Gould et al., 1985) nor did delays of up to 250 ms in the responsiveness of a tape-recorder to presses of a foot-pedal affect the amount of time it took for transcriptionists to complete a typing task (VanBalen and Eisler, 1989).

For other tasks, as long as the user is able to keep working or maintain their mental focus, even substantial delays have no effect on their ability to complete their given task. Grossberg et al. (1976) had subjects solve complex mathematical problems using a command line system. One command allowed the subjects to print relevant information. Grossberg inserted delays of up to 64 s in responding to this print command, though the subjects were able to continue using other commands to solve the mathematical problem during this delay. The length of these print delays had no effect on the subjects' overall time to solve the math problems. Treurniet et al. (1985) found that delays of nearly 30 s between reading text on screen and the presentation of a comprehension

question afterwards had no effect on the amount of time it took them to answer the questions.

It seems clear from the results of the research that as SRT increases user productivity is clearly affected. Over long periods of time, the number of tasks completed or commands issued to the computer decrease linearly as SRT increases. There is also evidence that increases in SRT affect the type of commands issued to the system and the overall user interaction with the system. When it comes to task completion time we see a progression of effect. When those delays interfere with the perceptual process, like for error rates, this causes significant disruption to the user and causes them to take longer to complete the task. If those delays do not interfere with perception, but just get in the way of completing the larger task, there is some evidence that there may be an optimal SRT, but these results are not clear. There is little effect when the user is able to go about his or her work. As in Grossberg's study, if the user can keep his or her train of thought or can keep making progress toward a larger goal, total task completion times can be unaffected even by large delays. Interestingly Schaefer and Kohlisch (1995) found that as delays increased, so did user performance time. However, *unexpectedly short* delays had an impact on performance while unexpectedly long delays did not. It seems that users can tolerate and adapt to delays so long as they do not interfere with their tasks or are at least as long or longer than expected.

2.3. User adjustment

Humans are an adaptable lot. We have shown throughout history that we can, and will, adjust to just about any environmental condition that exists, and that includes interactions with a computer system. In every study where some type of user adjustment as a result of changes in SRT was measured, a significant effect was found. Like the other categories mentioned above, user adjustment falls into two broad categories: user response time and user response style.

User response time (URT) is a measure of how long it takes a person to respond to the computer once the computer signals it is ready for more input. In the typical model of human–computer interaction, the human first issues some command to the computer, the computer takes some amount of time to respond to the command and once its work is completed, presents the user with some sort of command prompt, or otherwise appears ready for the next input, then the user responds with his or her next command. This URT has also been called, "user think time" and can be measured in a variety of ways.

The results of the research to date are abundantly clear: as SRT increases, so too does URT. When researchers were investigating employees in their normal working environment as computer programmers (Kosmatka, 1984; Lambert, 1984) they found clear, steady increases in URT with every increase in SRT. These effects were observed over thousands of commands issued over days of interaction with the computer system. Even in the laboratory, when the task simulated simple data entry (Butler, 1983, 1984; Martin and Corl, 1986; O'Donnell and Draper, 1996; Teal and Rudnicky, 1992; Williges and Williges, 1982) every study conducted found clear evidence that as SRT increased, so too did the amount of time it took users to begin typing in their next piece of data. There is, however, some evidence that the nature of the task plays a role as well. Martin and Corl (1986) had subjects perform a variety of tasks with a computer system. They found that for simple data entry, the relation between URT and SRT was as described above, but when the subjects were asked to solve more complex problems user response time tended to overshadow any delays in SRT. More complex problems required more thought time no matter how quickly the system was ready for the user.

Not only do users take longer to respond as SRT increases, but they also qualitatively change the kinds of commands they will issue to the computer once they do respond. As SRTs shorten, users adopt a more rapid, staccato style of issuing commands. As SRT increases, users tend to become more thoughtful, take longer to issue commands, and the commands they issue tend to be more complex and rich. For example, when performing text-editing tasks with full-screen text-editors, Gould et al. (1985) found that when subjects positioned the cursor with the arrow keys, in the short delay conditions, they tended to overshoot their targets and use more keystrokes than when they were in longer delay conditions. Guynes (1988) had subjects make a set of edits to a text document. Guynes found that subjects in the shorter delay conditions tended to use more commands to complete the edits than did subjects in the longer delay conditions. This is especially curious when recalling that subjects in that same study did not take any longer to complete the full set of edits in the long delay condition versus the short delay condition. When subjects were solving more complex problems such as issuing a series of commands to solve a complex mathematical problem (Grossberg et al., 1976), or solving an eights puzzle on a computer screen (O'Hara, 1994) they tend to issue fewer commands to solve the problem and the commands they issue tend to be more complex and more thoughtful according to the authors. Even under normal working conditions, when Kosmatka (1984) investigated the types of commands the programmers were issuing to the system, he found that the proportion of single-letter, more "trivial" commands increased as SRT decreased so that while it may have appeared that the programmers were more productive (they were issuing more commands) in reality they were just being less efficient.

The results of these studies show that users will pace themselves to the system. With a fast computer system, users respond quickly. As the system slows, so too will the pace of user interaction. As Dix (1992) noted pacing can be critical to the human–computer interaction. If the response time of the system does not match the pace of the type of interaction, then users become disrupted in their work and will have to adjust their actions to meet the speed of the system. One can therefore look at SRT as a kind of cost to the user. As SRT increases, so too does the cost of interacting with the computer. This increase in cost forces the user to make each command accomplish more and also provides time in which to develop strategies for parsimonious interaction. Under short SRT conditions, the cost of issuing a single command is trivial, therefore users feel free to issue more commands simply to see what will happen. As SRT increases, people become more parsimonious and tend to become more thoughtful before entering their next command. They want to make each command worth the cost. The study by O'Hara (1994) found direct evidence for this. He had subjects solve an 8s puzzle by issuing commands to a computer under three different conditions: in one, the cost to issue the command was varied by making the commands more complex and difficult to input; in another, the commands were simple, but the SRT was varied from fast to slow thereby imposing a cost on the user for each command input; and in the last, the manner in which the user had to undo previous commands was varied thereby imposing a different cost to the user for making mistakes. In all cases, the same pattern of behavior was observed: as the cost for making a move increased, users tended to solve the puzzle with fewer commands and to take longer before issuing the next command.

2.4. Psychological effects

A number of studies have been conducted that have measured the impact of SRT on various psychological variables. The typical measures include: satisfaction with the system, levels of annoy-

ance, irritation or boredom, subjective ratings of anxiety and the general perceived quality of the computer system or the work environment. Several researchers also investigated how various forms of feedback might mitigate some of these effects. The results are pretty clear cut: as SRT increases satisfaction decreases, dissatisfaction increases and users form a poorer opinion of the computer system. These effects, however, can be mitigated by providing certain types of feedback.

In general, as SRT increases user satisfaction with the computer system decreases, but the amount of delay a user will tolerate again depends on the nature of the task being performed. It takes very little delay in the response of an on-screen cursor (100 ms) (Gould et al., 1985) or a transcriptionist's foot-pedal (250 ms) (VanBalen and Eisler, 1989) for users to notice and become annoyed with the computer system. Furthermore, Gould found that users tended to believe that those delays affected their performance even when it did not.

When the tasks become more sequential and interactive such as in data entry, users tend to tolerate delays of several seconds before becoming annoyed with the system. Williges and Williges (1982) found that levels of annoyance increased significantly when the SRT exceeded 5 s and Schleifer and Amick (1989) found delays of 350 ms to be unnoticeable while variable delays ranging from 3 to 10 s significantly annoyed their typists. Similarly, when searching a 2D grid of numbers for a target (Thum et al., 1995) subjects were most annoyed when the delays between tasks was 4.5 s but they tolerated delays of .5 and 1.5 s.

As the interaction becomes more relaxed and self-paced, users' tolerance for delay seems to increase by an order of magnitude, increasing from seconds for sequential, interactive tasks to tens of seconds for browsing tasks. When reading articles presented on a computer screen and answering questions based on those articles (Planas and Treurniet, 1988; Treurniet et al., 1985) subjects did not become more significantly annoyed with the computer system until those delays exceeded about 15–20 s. In an online shopping task (Selvidge et al., 2000), delays of 30 or 60 s produced higher frustration than delays of 1 s though the subjects did not perceived the task as being any more difficult while Fischer et al. (2005) found steady decreases in user satisfaction as delay increased from 0 to 10 s. Bhatti et al. (2000) found that subject tolerance for going back to a previous web page to view groups of products decreased once SRT exceeded 11 s. There is such a clear relationship between levels of annoyance and irritation with the computer system and SRT that Planas and Treurniet (1988) even proposed an annoyance equation which described a linear relationship between annoyance levels and the square-root of the level of delay. However, it is also clear that the amount of delay tolerated changes with the type of task being performed.

Not only do user's levels of annoyance and frustration go up as SRT increases, but the perceived quality of the system is affected by the response time of that system as well. In their workplace studies, Barber et al. (1983) and Lambert (1984) consistently found their workers to have a more positive view of the computer system when the SRT was shorter than when it was longer. Bhatti et al. (2000) found that as web-page loading times increased, perceived quality of the subsequent page decreased. Hoxmeier and DiCesare (2000) found that after experiencing a delay of 12 s per page load, subjects were less likely to want to use the system again and were most likely to want to reuse the system when delays were set to zero. When it comes to simply following links on a web-page, both Ramsay et al. (1998) and Sears et al. (1997) found that as delays increased from about 2 s up to over 2 min, subjects consistently rated the subsequent page as being of poorer quality and of less interest.

Perhaps most interesting is the fact that if some type of feedback is provided as the user waits for the system to respond, most of the effects described above can be mitigated. Meyer et al. (1996) found that when subjects were presented with some type of on-screen feedback while waiting for the computer to respond to an action, their estimates of the amount of time they waited could be easily manipulated by the feedback. If the subjects were presented with a simple static, or blinking display, duration estimates were consistent and accurate. If they were presented with some type of incremental feedback in the form of changing text, a clock face with the hands revolving slowly or even an incrementally filling line of X's, this tended to affect their duration estimates. Users' perception of duration was longer when the feedback was updated more frequently than when the update was less frequent. Similar results were found by Planas and Treurniet (1988) when they found that subjects provided with feedback rated the delay as less annoying than subjects not provided with feedback. Bhatti et al. (2000) found that when web-pages loaded incrementally subjects tended to be more tolerant of higher levels of delay and rate the subsequent pages as having a higher quality than when the pages displayed nothing until the enforced delay was over. Finally, Nah (2004) found that subjects were less likely to abort loading a web-page when there was some type of feedback than when there was none. Clearly, the use of some types of feedback is important in mitigating the negative effects of delay in SRT.

It seems clear that increases in SRT leads directly to decreases in satisfaction with computer systems and increases in frustration, annoyance and irritation with the system. Unlike some of the productivity changes described above where there was oftentimes an optimal delay for specific tasks, changes in satisfaction or dissatisfaction with computer systems seems to stay relatively flat for some period of time and then change sharply and linearly with increases in SRT beyond a certain amount of time. The actual threshold of annoyance (the point at which users begin to become dissatisfied with the system) seems to vary considerably depending on the type of task. This threshold however can be manipulated through the use of feedback. Slower tempos of change in user feedback cause users to perceive less delay, which results in a higher threshold of annoyance.

2.5. Physiological effects

The final major area of study has been on how changes in SRT cause physiological changes in the users. Not many studies have been conducted in this area, but the ones completed are thorough, well-done and show clear and convincing evidence that changes in SRT cause direct changes in the physiological functioning of users. Researchers at the University of Wuppertal in Germany began systematically investigating the effects of SRT on user levels of stress and anxiety (Kuhmann et al., 1987; Kuhmann, 1989; Thum et al., 1995; Schaefer and Kohlisch, 1995; Kohlisch and Schaefer, 1996; Kohlisch and Kuhmann, 1997). They tested a variety of SRTs, controlled for the amount of work-load, mental stress and anxiety caused by the performance of the task, and have attempted to completely isolate the effect of system delay on user stress. In measuring stress and anxiety, they focused on physiologic measures such as heart rate, blood pressure and electrodermal skin conductance while also taking subjective reports of headaches and general feelings of discomfort. This group's research investigated one specific aspect of SRT: inter-task delay—that is, the amount of time between the completion of one task and the beginning of another, identical task. Generally, they have found that increases in inter-task delay result in more anxiety and that this increase in anxiety was not due to the stressful nature of the task to be performed. Furthermore, inter-task delays that are too short also increase anxiety in the subjects and cause more errors and greater dissatisfaction with the system. They have concluded that, for any given task, there is an optimal inter-task delay that will allow the user an

appropriate amount of time to prepare for the next task and to perform that next task with a minimum of errors and anxiety.

3. Existing SRT guidelines

Forty years of computer system response time research has led us to some general conclusions. Generally, faster is better. Users dislike delay and when computer systems do not respond quickly enough to their input, they become frustrated, angry, dissatisfied and exhibit more physiologic symptoms of stress regardless of the amount of work being done. Delays in the responsiveness of computer systems are most damaging when they interfere with atomic actions such as pressing keys and manipulating a mouse. Even minimal delays at this level cause significant disruption to the user and what is worse, as the complexity or difficulty of the task increases, the effect of such delays are magnified.

Once one moves beyond these low-level, atomic tasks the influence of SRT becomes less clear-cut. This is due, in part, to the natural human capacity to adapt to the environment or situation. Human–computer interaction is just that, an interaction. People, being adaptable, will adjust their pace, performance and work style in response to changes in SRT. When delay matches the user's readiness for task execution, all is well. As delays lengthen beyond this optimal, users will begin to change the pace at which they issue commands. They may take longer to respond to the system once it is ready, and they may even reorganize their commands or actions in an attempt to compensate for those delays. In the face of the longest delays, users will try to make each command count for more since the cost of issuing each command is so high. If the system is ready before the user is, users may again adjust their strategy and begin to issue simpler, more trivial commands. Their interaction style will become more staccato and they may adopt a trial and error approach. In either case, such changes in strategy can offset changes in SRT so that overall task completion time is unaffected.

Using these basic conclusions, Shneiderman, in 1987, proposed a four-tier, task-focused model of suggested computer SRTs (Shneiderman, 1987) that remains unchanged to this day (Shneiderman et al., 2009). The limits suggested by that model are given in Table 1 below.

In his recommended SRT guidelines, Shneiderman takes a task-focused approach. He states that for simple tasks, such as typing and moving the mouse, the system must respond essentially as quickly as possible. However, as task complexity increases, users will tolerate or prefer the down-time imposed by system delays to recover from the task just completed and to plan the next course of action. It is the complexity of the task that needs to be considered in determining the appropriate range of system response times.

To his credit, Shneiderman asks more questions than the available research or his model are able to answer. He points out that additional research is needed to determine how specific types of tasks can be assigned to his various categories. He suggests that system administrators monitor user performance speed and error rates for various tasks and then tune system response time to produce the best productivity with the fewest errors. Using his guidelines, system designers can know roughly when to provide feedback about possible delays so that users can adjust their behavior and avoid frustration with the system. He freely acknowledges that much work is needed to further understand the interactions between system response time and user productivity.

However, there are other problems with Sheiderman's model. His model suggests a graduated response time based on task complexity. The model considers key-presses and mouse-clicks to be the simplest tasks and suggests increasing response times as task complexity increases. However, are not even the most complex tasks with a computer carried out via key-presses or mouse-clicks? What characteristics make one task simple and another complex? Is complexity determined simply by the number of key-presses or mouse-clicks required or does complexity derive from the thinking effort required of the user? Again, Shneiderman asks many of these same questions himself, but his response-time guidelines provide no framework around which to build or ask these questions.

Looking at the same historical research, Seow had a different view of the human–computer interaction (Seow, 2008). In his 2008 book on the psychology of time perception in software he also proposed a four-tier model of suggested SRTs however he took a more user-centric approach. Rather than tying response time guidelines to specific actions or task complexity, he proposed that interactions with a computer can be assigned to categories based on user expectancies. Limits of tolerance for delay would then depend on the kind of interaction and expectation a user has for that type of task rather than the task's complexity. His limits of acceptable response times are given in Table 2 below.

The *instantaneous* category covers low-level actions such as key-presses and mouse-clicks. Like Shneiderman, he suggests that delays of longer than 100–200 ms will be disruptive to users. An *immediate* response is called for in situations where the computer is expected to acknowledge an action taken by a user, such as a screen tap or mouse click to view the next page of a document. The computer may not need to complete the action within that time frame, but it must begin responding and convey this fact to the user. At the next level, Seow describes *continuous* interactions with a computer. In these cases, computers need to respond within his suggested 2–5 s so as to maintain the flow of the interaction. If the computer takes too long to respond and does not provide any feedback, the user may begin to think there is a problem with the system. The final category, called *captive*, covers those tasks for which the user must simply wait for results, but will eventually give up if no response occurs. In these situations, Seow suggests that providing some type of feedback within 7–10 s will keep the user from abandoning the task all together.

Seow's model, like that of Miller, treats human–computer interaction as a conversation for which a certain pace must be maintained. If certain timings are not observed, the conversation will fail. Seow's model is an improvement over Shneiderman's because it incorporates the interactive nature of tasks with computers and because it provides a better means for assigning an interaction to a category. But ignoring the nature of the task also seems wrong. One can imagine playing chess against a computer opponent. At high levels of difficulty, system delays of more than 10 s per move would not be uncommon. Is the human likely to abandon his game

Table 1
Shneiderman's response-time categorization.

Task	Response time
Typing, mouse movement	50–150 ms
Simple frequent tasks	1 s
Common tasks	2–4 s
Complex tasks	8–12 s

Table 2
Seow's response-time categorization.

Expectation	Response time
Instantaneous	100–200 ms
Immediate	.5–1 s
Continuous	2–5 s
Captive	7–10 s

because he thinks the computer is unresponsive or are such delays perfectly acceptable because of the complexity of the task at hand? In a two-human chess game, is not this aspect of waiting for the other player an almost essential part of the experience? Shneiderman's model acknowledges that tasks can have differing levels of complexity and therefore require different levels of responsiveness whereas Seow's model avoids taking this into account.

To summarize, Shneiderman and Seow take different approaches to determining optimal response times. Shneiderman's model is task-focused, with suggested response times that increase with task complexity. Its weaknesses are that it does not consider the user's expectations and that it does not specify how to determine task complexity. Seow's model emphasizes user expectations in a conversational interaction and provides categories of interaction that seem easier for developers to determine. Both models present four categories of response time, but these are loosely defined and appear to be broad categorizations placed over a continuous range of values without clear cut points. Neither model does an especially good job of explaining the results of prior research except at the level of instantaneous, mouse/key-oriented interaction. What is needed is a model that takes both task-oriented and user-oriented concerns into account and provides a clear direction for future research.

4. Categorizing tasks for SRT: control versus conversation

Past research on system response time is abundantly clear on one point: for low-level tasks such as key-presses and mouse-clicks, the computer system must respond as fast as possible. Any perceptible delays will be very disruptive to the user. Beyond this, the literature review lacked clear-cut lessons. But perhaps the lack of results reflects a flaw in our model of human–computer interactions. Perhaps the point is that low-level, key-press and mouse-click tasks are really a different category of interaction. Perhaps when Miller wrote his paper about "man computer conversational transactions," he was wrong to include the lowest level of interaction in his taxonomy Miller (1968). Hence, we propose that there are two categories of interaction with a computer: control tasks and conversational tasks.

4.1. Control tasks

Control tasks are those for which the user would like the software to behave like a physical device directly controlled through key-presses, mouse-clicks or screen-taps. In a control task, the user expects to see a one-to-one correspondence between her actions and the system's output. The user is relying principally on her perceptual and motor coordination skills, which are the parts of mental processing that are most efficient. As a result, the user does not need extensive "think time" to determine what action to take next. So for control tasks, the computer should appear to respond instantly, as if it has no need to reason about how to respond to the user's actions. The optimum SRT will be zero, while delays above 200 ms will be readily perceived and will likely disrupt the user's work. In control tasks, the computer should essentially vanish, giving the user the illusion of direct control over a physical device.

Of course, this is the essence of the direct manipulation user interface experience and the best software tools on today's touch-screen tablets exemplify this quality. On a modern tablet, the user can turn the pages of an electronic book by simple gestures that evoke metaphors of turning physical pages. The system hides the fact that the computer is doing extensive processing to perform these operations. When the computer cannot hide its processing effort for some task, then the metaphor is weakened and the interaction starts to appear conversational, even if the user would prefer that it not be. For example, a photographer editing a digital photo should not feel as if she is issuing commands to the computer, which then applies an effect and displays the result. Rather, she should feel as if she is directly controlling the photo until she has produced the desired results. If a user is writing a paper, he should be able to focus on the writing and the computer should display what he writes as quickly as possible.

Notice that improvements in computing power can change which control tasks can be supported effectively. For example, in the early days of personal computing, preparing a paper for publication typically involved typing the words and various layout commands into a file, saving that file, running a typesetting program, viewing the results, fixing the errors and repeating the process. Modern word-processing software now handles the majority of this workflow automatically while the user is typing. So, what was previously handled as a complex conversational interaction is now largely supported as a control task. One can imagine the same thing becoming true for web browsing or other internet-based applications. As internet access speeds increase, users may begin to feel as if they are simply viewing interactive documents that respond to the touch of a finger, regardless of where the documents are located.

The tasks for computer scientists and software engineers are to fine tune the response time guidelines and to continually evaluate various tasks to determine when they can or should move into the realm of control tasks. Combining Shneiderman's and Seow's guidelines, response times of 50–200 ms are necessary for a user to feel as if he is controlling the device. These numbers have sound experimental footings, but the range is fairly large. Which types of interactions require a 50 ms response time and which require 200? Do key-presses and mouse-clicks have the same delay detection thresholds as touch-screen devices? Humans display a great amount of variability in their perceptual capabilities, so what proportions of the population are covered by these ranges? If the delay is 100 ms, will 50% of the people notice a delay or will 10% notice? How about at 200 ms? While likely only of academic interest, it would be nice to have firm, experimentally established response-time curves for different types of interactions that show what percentage of the population would notice a delay at various delay levels. Finally, software engineers need to be aware of how increases in computing power can support new interaction styles so that they can recognize when a task is moving into the control realm.

4.2. Conversational tasks

Conversational tasks are the second broad class of computer interaction, relative to SRT. In a conversational task, the user is still controlling the computer, but the interaction is not focused on control per se, but rather on some larger task at hand. The user issues commands to the computer, waits for results, thinks about the results to some degree, and issues more commands based on the response. We believe that it is for conversational tasks that the SRT research results are inconsistent and that neither Shneiderman's or Seow's classification systems are truly helpful. Shneiderman's system does not explicitly take this conversational nature into account at all, focusing instead on task complexity. Seow attempts to use this conversational nature of human–computer interaction in defining these interactions in terms of user expectancies, but the classification of expectancies is lacking.

Consider a simple example where a user must enter 100 12-digit serial numbers into a computer for order processing. Each serial number can include both digits and characters (`A43B-9972–382Z`,

for example) and each four-digit chunk is being entered into a separate field on a database form, hence the user must tab between those fields. When done entering a single serial number, the user will press enter to submit the number and move onto the next. Now in this scenario, what exactly is the task and how should it be categorized?

The task could be any of the following:

- Entering all 100 12-digit serial numbers.
- Entering a single serial number (repeated 100 times).
- Entering a chunk of 4 characters (repeated 300 times with different "go to next step" actions, depending on context).
- Entering single characters, including tab and enter.

Suppose the task is to enter 100 serial numbers. Intuitively, it seems likely that users would like to pause for a substantial period between the blocks of 100 numbers. But it is not clear that this task is "complex" when using Shneiderman's classification since the user's actions are so simple and repetitive. Seow's classification system does not seem to address this scenario at all. One would hardly call this a captive task, since it is unlikely the user will give up waiting for the computer to respond with the next list, and yet, a lengthy delay on the part of the computer system might be desired by the user and may even increase their performance on the next list.

Suppose, instead, that the task is entering a serial number, but the task is repeated 100 times. Again using intuition, this could easily fit Shneiderman's simple or common task categories or Seow's categories of immediate or continuous response. In either case, delays by the computer of 1–5 s might be tolerated by the user without it having an undue influence on his or her performance. But since those delays are part of the larger task of entering all 100 serial numbers, the addition of those delays could make that larger task more difficult for the user and cause additional stress and anxiety and affecting performance. Effects such as this have been described in the literature.

This phenomenon of lower-level delays affecting higher-level tasks can also occur at the four-digit-chunk and single character level. We believe that the chunk level would be considered simple by Shneiderman and that Seow would say that users should expect an immediate response, while the single character level would fall at the lowest level under both models.

Of course, this model of tasks underlies the keystroke level analysis of Card et al. (1983). As they suggest, every task with a computer is recursively composed of sub-tasks which can eventually be broken down to the keystroke (or mouse-click) level. As can be seen from our example, delays in computer SRT can occur at any of those sub-task boundaries but what is of paramount importance here is the pace of the interaction. The literature has shown that users will adapt their pace to the responsiveness of the system, but as Dix (1992) pointed out the delays cannot be so great as to interfere with an acceptable flow of the conversation. Hence, in a conversational task, delays can occur in various places within that conversation and as long as they do not interfere with the pacing of the conversation, the user should be able to complete the task with a minimum of errors at a maximally productive rate. If the delays do not match the pacing of the conversation, disruption will occur. How much delay is needed or tolerable will depend on the complexity of the sub-task completed just prior to the delay.

The question then is how to measure task or sub-task complexity? The effort required to complete a given task with a computer is composed of two parts: the physical load and the cognitive load. In conventional computer interactions, the physical load can best be thought of as the number of key-presses, mouse-clicks or finger-flicks required to complete the task. Typing an entire paragraph has much higher physical load than a single button click. Cognitive load is the amount of mental effort necessary to understand and process the information being presented and to then formulate and carry out a response. User knowledge and experience can affect cognitive load and the task itself can have an inherent level of mental effort required for comprehension and reasoning. Physical load and cognitive load can vary independently. For example, typing in a lengthy sequence of simple words would require much physical effort but very little cognitive effort for an experienced typist. Choosing and executing a move in a computer chess games would require very little physical effort, but might require a significant amount of cognitive effort.

As per Shneiderman, when a user completes a sub-task that is more complex (i.e., has a higher combination of mental and physical load) he will tolerate or even prefer a longer delay from the system. This delay will help maintain an acceptable conversation pace and will match the user's expectations for such a conversation, as suggested by Seow. This pacing will then produce the greatest performance from the user with the fewest errors and the highest satisfaction with the system.

Clearly this part of our proposed framework will require validation, but it provides many fruitful avenues for research. Identifying sub-task boundaries would be one such avenue. Several studies have been done that have examined how *interruptions* affect user satisfaction and productivity especially when those interruptions occur along sub-task boundaries (Bailey et al., 2000, 2001; Adamczyk and Bailey, 2004). Similar procedures could be used to address how *delays* at those same boundaries affect performance. Additionally, ways to measure the mental and physical load of sub-tasks need to be identified. The NASA-TLX (Hart and Staveland, 1988) is one such measure. It has been shown to be a reliable measure of the amount of "work" a user feels was needed to complete a task. Other similar measures exist (Neerincx, 2003). Next would come the task of validating Shneiderman's limits of system response time and determining how they are related to the measured task complexity. It is quite possible that his dichotomy of simple versus complex may suffice to classify the majority of computer interaction tasks, but that remains to be seen. Finally, it would be of interest to see how the addition of delays beyond those tolerable by the user affect the overall perceived difficulty of the task. If the delays interfere with the desired pacing of the task, is the task viewed as being more difficult or is the difficulty unaffected but frustration and annoyance increase? Clearly, there is a significant amount of work to be done in this area.

Given that computers may have finally reached where they are "fast enough" for most tasks that most users perform, software engineers may be called upon to bring more tasks into the control realm and have the software behave more like a physical device. Researchers should continue to pursue the precise limits of delay detection for the full range of interaction types so that the engineers have hard numbers against which to measure the performance of their applications. There will also remain some computer tasks that are more conversational in nature. For these tasks, researchers need to identify boundaries at which delays can or should be inserted so as to make the software easier to interact with. They also need to determine the precise limits of those boundaries so that engineers can fine-tune their applications to match those limits. Processor-intensive tasks could be placed at these boundaries to reduce the chance that users would notice the delays. We are at a stage where researchers and software engineers can work hand-in-hand to make the software we use fit the expected, natural flow of interaction with a computer. Sometimes that natural flow will give the user a sense of direct control and other times that flow may be more conversational.

5. Final words

Forty years of research into acceptable limits of computer system response time during a human–computer interaction has produced a wealth of data, but few clear guidelines. Prior classification systems such as the ones proposed by Shneiderman and Seow have gone a long way toward putting these results into a framework on which future research can be built, but both ultimately fail to cover the full range of human–computer interactions. We proposed a new categorization system for looking at human–computer interaction that attempts to fuse both Shneiderman's task-centric and Seow's expectancy-centric systems into a more coherent framework. We suggest that all human–computer interactions can be viewed as falling into one of two broad categories: control tasks or conversational tasks. For control tasks, instantaneous response times are needed whereas for conversational tasks, some delay may be preferable depending on both the complexity of the task and the user expectancies of the conversational nature of the task. We hope that this categorization can spur future research into this area so as to provide better guidelines for practitioners and researchers in this area.

References

Adamczyk, P.D., Bailey, B.P., 2004. If not now, when? The effects of interruption at different moments within task execution. In: Dykstra-Erickson, E., Tscheligi, M. (Eds.), Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, vol. 6. ACM, pp. 271–278 (ACM).

Bailey, B.P., Konstan, J.A., Carlis, J.V., 2000. Measuring the effects of interruptions on task performance in the user interface. In: IEEE International Conference on Systems, Man and Cybernetics, IEEE, pp. 757–762.

Bailey, B.P., Konstan, J.A., Carlis, J.V., 2001. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: The Proceedings of INTERACT 2001, pp. 593–601.

Barber, R.E., Henry, J., Lucas, C., 1983. System response time, operatory productivity, and job satisfaction. Communications of the ACM 26 (11), 972–986.

Bhatti, N., Bouch, A., Kuchinsky, A., 2000. Integrating user-perceived quality into web server design. In: Proceedings of the 9th International World-Wide Web Conference. Elsevier, pp. 1–16.

Butler, T.W., 1983. Computer response time and user performance. In: ACM (Ed.), CHI '83 Proceedings, 1983.

Butler, T.W., 1984. Computer response time and user performance during data entry. AT & T Bell Laboratories Technical Journal 63 (3), 1007–1017.

Card, S.K., Moran, T.P., Newell, A., 1983. The Psychology of Human Computer Interaction. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Dannenbring, G.L., 1984. System response time and user performance. IEEE Transactions on Systems, Man and Cybernetics SMC-14 (3), 473–478.

Dix, A., 1992. Pace and interaction. In: Monk, A., Diaper, D., Harrison, M. (Eds.), Proceedings of HCI '92: People and Computers VIII, pp. 193–207.

Fischer, A.R.H., Blommaert, F.J.J., Midden, C.J.H., 2005. Monitoring and evaluation of time delay. International Journal of Human–Computer Interaction 19 (2), 163–180.

Goodman, T.J., Spence, R., 1978. The effect of system response time on interactive computer aided problem solving. In: Proceedings of the SIGGRAPH '78 Conference. ACM, New York, pp. 100–104.

Goodman, T.J., Spence, R., 1981. The effect of computer system response time variability on interactive graphical problem solving. IEEE Transactions on Systems, Man and Cybernetics 11 (3), 207–216.

Goodman, T., Spence, R., 1982. The effects of potentiometer dimensionality, system response time, and time of day on interactive graphical problem solving. Human Factors 24 (4), 437–456.

Gould, J.D., Lewis, C., Barnes, V., 1985. Cursor movement during text editing. ACM Transactions on Office Information Systems 3 (1), 22–34.

Grossberg, M., Wiesen, R.A., Yntema, D.B., 1976. An experiment on problem solving with delayed computer responses. IEEE Transactions on Systems, Man and Cybernetics, 219–222.

Guynes, J.L., 1988. Impact of system response time on state anxiety. Communications of the ACM 31 (3), 342–347.

Hart, S.G., Staveland, L.E., 1998. Human Mental Workload, North-Holland, pp. 239–250 (Chapter Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research).

Hoxmeier, J.A., DiCesare, C., 2000. System response time and user satisfaction: an experimental study of browser-base applications. In: Proceedings of the Americas Conference on Information Systems, pp. 140–145.

Kohlisch, O., Kuhmann, W., 1997. System response time and readiness for task execution – the optimal duration of inter-task delays. Ergonomics 40 (3), 265–280.

Kohlisch, O., Schaefer, F., 1996. Physiological changes during computer tasks: responses to mental load or to motor demands? Ergonomics 39 (2), 213–224.

Kosmatka, L.J., 1984. A user challenges value of subsecond response time. Computerworld, ID/1–ID/18.

Kuhmann, W., 1989. Experimental investigation of stress-inducing properties of system response times. Ergonomics 32 (3), 271–280.

Kuhmann, W., Boucsein, W., Schaefer, F., Alexander, J., 1987. Experimental investigation of psychophysiological stress-reactions induced by different system response times in human–computer interaction. Ergonomics 30 (6), 933–943.

Lambert, G.N., 1984. A comparative study of system response time on program developer productivity. IBM Systems Journal 23 (1), 36–43.

MacKenzie, I.S., Ware, C., 1993. Lag as a determinant of human performance in interactive systems. In: Proceedings of the ACM Conference on Human Factors in Computing Systems – INTERCHI '93. ACM, New York, pp. 488–493.

Martin, G.L., Corl, K.G., 1986. System response time effects on user productivity. Behaviour and Information Technology 5 (1), 3–13.

McCain, J.E., 1993. The effects of system response time on directory assistance operator worktime. In: Luczak, H., Cakir, A., Cakir, G. (Eds.), Work with Display Units 92. Elsevier Science B.V, pp. 210–214.

Meyer, J., Shinar, D., Bitan, Y., Leiser, D., 1996. Duration estimates and users' preferences in human–computer interaction. Ergonomics 39 (1), 46–60.

Miller, R.B., 1968. Response time in man–computer conversational transactions. In: Proceedings Spring Joint Computer Conference 1968, vol. 33. AFIPS Press, Montvale, NJ, pp. 267–277.

Nah, F.F.-H., 2004. A study on tolerable waiting time: how long are web users willing to wait? Behaviour and Information Technology 23 (3), 153–163.

Neerincx, M.A., 2003. Handbook of Cognitive Task Design, Human Factors and Ergonomics, Routledge, USA, pp. 283–305 (Chapter 13).

O'Donnell, P., Draper, S.W., 1996. How machine delays change user strategies. ACM SIGCHI Bulletin 28 (2), 39–42.

O'Hara, K., 1994. Cost of operations affects planfulness of problem-solving behaviour. In: Plaisant, C. (Ed.), Conference Companion on Human Factors in Computing Systems. ACM Press, New York, NY, pp. 105–106.

Planas, M.A., Treurniet, W.C., 1988. The effects of feedback during delays in simulated teletext reception. Behaviour and Information Technology 7 (2), 183–191.

Ramsay, J., Barbesi, A., Preece, J., 1998. A psychological investigation of long retrieval times on the world wide web. Interacting with Computers 10, 77–86.

Schaefer, F., 1990. The effect of system response times in temporal predictability of work-flow in human–computer interaction. Human Performance 3 (3), 176–183.

Schaefer, F., Kohlisch, O., 1995. The effect of anticipatory mismatch in work flow on task performance and event related brain potentials. In: Grieco, A., Molteni, G., Piccoli, B., Occhipinti, E. (Eds.), Work with Display Units 94. Elsevier Science B.V., pp. 241–245.

Schleifer, L.M., Amick, B.C., 1989. System response time and method of pay: stress effects in computer-based tasks. International Journal of Human Computer Interaction 1 (1), 23–29.

Sears, A., Jacko, J.A., Borella, M.S., 1997. Internet delay effects: How users perceive quality, organization and ease of use of information. In: CHI '97 Extended Abstracts on Human Factors in Computing Systems. ACM Press, New York, NY, pp. 353–354.

Selvidge, P.R., Chaparro, B., Bender, G.T., 2000. The world wide wait: effects of delays on user performance. In: Proceedings of the IEA 2000/HFES 2000 Congress, vol. 1, pp. 416–419.

Seow, S.C., 2008. Designing and Engineering Time. Addison-Wesley.

Shneiderman, B., 1987. Designing the User Interface: Strategies for Effective Human–Computer Interaction, first ed. Addison-Wesley.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., 2009. Designing the User Interface: Strategies for Effective Human–Computer Interaction, fifth ed. Addison-Wesley.

Teal, S.L., Rudnicky, A.I., 1992. A performance model of system delay and user strategy selection. In: CHI '92 Extended Abstracts on Human Factors in Computing Systems. ACM Press, New York, NY, pp. 295–305.

Thum, M., Boucsein, W., Kuhmann, W., Ray, W.J., 1995. Standardized task strain and system response times in human–computer interaction. Ergonomics 38 (7), 1342–1351.

Treurniet, W.C., Hearty, P.J., Planas, M.A., 1985. Viewers' responses to delays in simulated teletext reception. Behaviour and Information Technology 4 (3), 177–188.

VanBalen, P.M., Eisler, L.R., 1989. Evaluation of audio response time delay requirements for digitized audio. In: Proceedings of the Human Factors Society 33rd Annual Meeting, vol. 1, pp. 234–238.

Williges, R.C., Williges, B.H., 1982. Modeling the human operator in computer-based data entry. Human Factors 24 (3), 285–299.