Int. J. Human-Computer Studies 72 (2014) 358–365

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

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

Predictivity of system delays shortens human response time ☆

Roland Thomaschke [a,b,*], Carola Haering [b]

[a] Institut für Psychologie, Universität Regensburg, Universitätstraße 31, 93053 Regensburg, Germany
[b] Institut für Psychologie, Universität Würzburg, Röntgenring 11, 97070 Würzburg, Germany

ARTICLE INFO

Article history:
Received 3 September 2012
Received in revised form
23 October 2013
Accepted 6 December 2013
Communicated by P. Mulholland
Available online 16 December 2013

Keywords:
System delay
Human computer interaction
Predictivity
Predictability

ABSTRACT

System delays considerably affect users' experience and performance. Research on the psychological effects of system delays has focused on delay length and variability. We introduce delay predictivity as a new factor profoundly affecting user performance. A system delay is predictive when its duration is informative about the nature of consecutive interaction events. We report an experiment ($N=122$) where short delays were differently distributed across two alternative target stimuli in a choice response task. We manipulated variability and predictivity of delays. For one group of participants the delays were of constant duration. For three other groups the delays were variable, but differed in predictivity. They were either non-predictive, probabilistically predictive (they predicted the targets with a probability of 0.8), or deterministically predictive. Performance with constant delays was superior to performance with variable non-predictive or with probabilistically predictive delays. Surprisingly, participants with deterministically predictive delays outperformed participants in all other groups. This has important implications for interface design, whenever there is some degree of freedom in scheduling system delays. Best performance is achieved with predictive delays, but only when deterministic predictivity can be achieved. Otherwise, constant delays are to be preferred over variable ones.

1. Introduction

When interacting with a computer, users often encounter waiting times between their input and the computer's response. These delays are commonly referred to as system delays (Selvidge et al., 2002; Szameitat et al., 2009) or system response times (Dabrowski and Munson, 2011; Schleifer and Amick, 1989). System delays are, on one hand, caused by constant properties of the system such as processing speed, network bandwidth or the complexity of the requested computation. On the other hand a number of transient factors influence system delays, such as network congestion, background processes, or a variety of other factors (Seow, 2008). Research on Human Computer Interaction (HCI) has shown that system delays can enormously influence users' experience and performance (Ceaparu et al., 2004; Nah, 2004; Thum et al., 1995). Although, due to a tremendous increase in computational processing speed, system delays are negligible in some contemporary HCI interfaces, they are still a major cause for users' discomfort and low performance in others (e.g., the Internet, see also Rose et al., 2009; Seneler et al., 2009). Many recent studies have investigated how the negative effects of delays can be managed, or (if possible) avoided by interface design (Branaghan and Sanchez, 2009; Galletta et al., 2006; Krejcar, 2009).

Two important factors determining the effects of delays on users' experience and performance are the delays' lengths, and their variability (Kuhmann, 1989; Kuhmann et al., 1987; Schaefer, 1990). Before introducing a third factor – predictivity – we briefly review previous literature on delay length and variability.

1.1. The length of system delays

There is an almost universal consensus in the literature that long waiting times are detrimental to users' performance and satisfaction (Martin and Corl, 1986; Schaefer, 1990; Seow, 2008; Simoens et al., 2011). Particularly, long waiting time in internet applications do considerably affect performance and lead to user frustration. Thus, loading time is a major issue in quality of service in the context of internet applications (Liaw and Huang, 2006). Even in domains with much shorter delays, like computer games, delays have been shown to negatively affect performance (Szameitat et al., 2009). Occasionally, performance improvements by lengthening of delays have been described (e.g., Barber et al., 1983; Sellier and Chattopadhyay, 2009). These instances seem, however, to be restricted to contexts where duration of a process signals trustworthiness, like, for example, online-payment mechanisms.

☆ This paper has been recommended for acceptance by P. Mulholland.

* Corresponding author at: Institut für Psychologie, Universität Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany. Tel.: +49 941 943 3776; fax: +49 941 943 1995.

E-mail addresses: Roland.Thomaschke@psychologie.uni-regensburg.de (R. Thomaschke), haering@psychologie.uni-wuerzburg.de (C. Haering).

1071-5819/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijhcs.2013.12.004

Thus, designers should, if possible, reduce the length of system delays, in order to increase user performance and satisfaction. These findings are corroborated by numerous studies in cognitive psychology, showing that response times to target stimuli increase with the length of preceding warning intervals of constant duration, except for very short (<300 ms) intervals (e.g., Leonhard et al., 2012; see Los and Schut, 2008; Müller-Gethmann et al., 2003, for reviews). These studies typically apply the *foreperiod paradigm* (Niemi and Näätänen, 1981). In this paradigm a target stimulus is preceded by a task irrelevant warning stimulus. The duration between warning and target stimulus – referred to as *foreperiod* – systematically affects performance (Rolke, 2008; Rolke et al., 2007; Rolke and Hofmann, 2007; Seibold et al., 2011; Seibold et al., 2011). However, when durations vary randomly between trials, performance *increases* with foreperiod duration (Los and Horoufchin, 2011; Steinborn and Langner, 2011; Steinborn et al., 2010; Steinborn et al., 2008; Steinborn et al., 2009). Yet, overall, responses are on average slower for variable than for constant foreperiods (Cardoso-Leite et al., 2009; Los et al., 2001; Mattes and Ulrich, 1997).

1.2. Variability of delays

The findings concerning behavioral effects from variable foreperiods in basic cognitive psychology have been confirmed in applied research with human–computer interfaces. The variability of delays the user encounters in human–machine interaction can considerably affect performance and satisfaction. System delays are referred to as *constant* when all system responses follow the preceding user input after one and the same time interval. System delays are referred to as *variable,* when the user is confronted with more than one possible delay duration. Variability can come in different degrees. Roast (1998) has defined the degree of variability as the span between the shortest and the longest possible interval duration (see also Fischer et al., 2005). It is a well-established finding in basic human performance research that choice responses are on average faster after constant than after variable delays (Cardoso-Leite et al., 2009; Wundt, 1874, see above). Since the early days of ergonomic research, this finding has been validated in several studies in human–machine interaction (Awramoff, 1903; Weber et al., 2013). HCI research has shown that increased variability has detrimental effects on user satisfaction (Fischer et al., 2005) and performance (Weber et al., 2013). Weber et al. (2013), for example, manipulated delay variability in an E-Mail program. Users' response latencies significantly increased with increasing delay variability.

1.2.1. Reducing variability: shortening and lengthening

As delay variability has negative effects on user experience and performance, designers should attempt to minimize variability in HCI interfaces. There are several ways to reduce variability. One option is obviously the reduction of extraordinarily long delays, like, for example, internet-download times. In order to reduce variability, such reduction must be specific to long delays, in the sense that already short delays are not also shortened (Roast, 1998). As the delays are caused by factors inherent in the system, delay reduction requires some kind of technical optimization of the computational process that causes long delays. Such technical optimizations, however, are beyond the scope of HCI interface design, and are, thus, not the focus of the present study.

A technically much less demanding way of reducing variability is the selective lengthening of short delays. This is the approach taken by Weber et al. (2013). In that study, for one group of participants, interaction with an email program was unpredictably interrupted by delays of 7 different durations. In another group, 5 of the possible durations were lengthened in a way that all delays could now have only 2 different lengths. The selective lengthening of delays considerably improved participants' performance. Likewise, Sellier and Chattopadhyay (2009) suggested to selectively add delays to unusually short web-page loading times to avoid the impression that “something is not right” with the web pages. Selective lengthening has the advantage that no sophisticated technical improvement is required. No computational process must be optimized. Delays must only be added in appropriate places.

It has, however, the major disadvantage that it also prolongs the total interaction time with the system. This issue becomes particularly problematic with regard to user performance and user frustration. An important reason to reduce variability is to improve user performance in the sense of speeding up users' responses (see Szameitat et al., 2009; Weber et al., 2013). It is, however, unlikely that the delays added to reduce variability will be compensated by the times saved by shorter user response latencies (though there are presently no systematic investigations on this issue). However, in Weber et al.'s study, due to the longer delays, total interaction time was longer in the low variability condition, although users' response latencies were reduced. Nevertheless, user satisfaction was not decreased in the condition with on average longer waiting times.

1.2.2. Reducing variability: scheduling

Another means for reducing variability is scheduling of delays. As described above, changing the obtainable system speed per se is beyond the scope of interaction design. However, scheduling enables interaction designers to speed up system response times by optimizing the use of processing power (Blazewicz et al., 2007). Scheduling requires that there are at least some degrees of freedom concerning the point in time when an interaction has to take place during the computational process. Scheduling is obviously not possible when the processing capacities of the system are at any time exclusively devoted to processing one input of one individual user. This is, however, the model implicitly or explicitly assumed by most traditional models informing temporal variability research in HCI (e.g., Roast, 1998). Consequently, scheduling has not been considered as an option to reduce variability.

Most modern computer systems are, however, not covered by those models. Due to the growing application of parallel computing, it is often the case that different processes or different users share a single processor or a set of processors. In such scenarios the need for some kind of scheduling emerges (Szameitat et al., 2009). The interface designer has some degree of choice how to distribute processing time over interaction events.

For example, in many programs' download and installation procedures, dialogs with the user are scheduled parallel to the download. Users provide information about the installation path, program settings etc. while the program is already downloading. This renders the system's delays less variable compared to situations with one long delay during the download and several almost instantaneous dialog interactions before or after (see Seow, 2008). Another example is an algorithm, developed by Pons (2006), which reschedules processing capacity from fast loading to long loading web pages, in order to reduce delay variability.

Scheduling combines the advantages of shortening and lengthening delays, which were discussed above. On the one hand, it makes system delays less variable without making computational processing technically faster. On the other hand, rescheduling avoids adding empty delays during which no processing takes place. Thus, variability can be reduced without artificially lengthening the total interaction time. Scheduling allows a system designer to homogenize intervals (e.g., by separating long delays and uniting short ones), and to also manipulate regularity between delays and interaction-events.