HAL open science

Measurement of Remote Response Delay in Multi-Synchronous Collaborative Editing

Claudia-Lavinia Ignat, Gérald Oster, Meagan Newman, Valerie Shalin, François Charoy

▶ **To cite this version:**

Claudia-Lavinia Ignat, Gérald Oster, Meagan Newman, Valerie Shalin, François Charoy. Measurement of Remote Response Delay in Multi-Synchronous Collaborative Editing. [Research Report] RR-8419, INRIA. 2013, pp.20. hal-00917317

HAL Id: hal-00917317

https://inria.hal.science/hal-00917317

Submitted on 11 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Measurement of Remote Response Delay in Multi-Synchronous Collaborative Editing

Claudia-Lavinia Ignat, Gérald Oster, Meagan Newman, Valerie Shalin, François Charoy

RESEARCH REPORT
N° 8419
December 2013
Project-Team Score

ISSN 0249-6399 ISRN INRIA/RR--8419--FR+ENG

Measurement of Remote Response Delay in Multi-Synchronous Collaborative Editing

Claudia-Lavinia Ignat*, Gérald Oster†, Meagan Newman‡, Valerie Shalin§, François Charoy†

Project-Team Score

Research Report n° 8419 — December 2013 — 20 pages

Abstract: In this study we examine the performance consequences of simulated network delay in collaborative document editing. Related studies suggest that while delay in the distribution of an individual's work to the team is a potential influence on performance, the impact is a function of strategy and task. However, a dearth of quantitative research in the domain of document editing makes it difficult to evaluate either concern for delay or the efficacy of compensatory strategies. The present study measures performance on an artificial document editing task with a time constant and metrics for process and outcome suitable for experimental study. Results suggest that strategy in the distribution of work influences task outcome at least as much as delay in the distribution of work in progress. However, a paradoxical interaction between delay and strategy emerged, in which the more generally effective, but highly coupled strategy was also more sensitive to delay.

Key-words: Collaborative editing, groupware, delay, usability measurement

This work is partially funded by the USCoast Inria Associated Team

* Inria, Université de Lorraine, CNRS
† Université de Lorraine, Inria, CNRS
‡ Department of Psychology, Wright State University
§ Department of Psychology, Wright State University, Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis)

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Étude de l'impact du temps de réponse distant dans un environnement d'édition collaboratif multi-synchrone

Résumé : Ce travail porte sur l'étude des performances de collaborateurs, rédigeant un document partagé de manière collaborative en temps réel, face à un délai simulé sur le média de communication réseau.

Mots-clés : Édition collaborative, outils de travail de groupe, délai, utilisabilité, mesure

1 Introduction

Computer science work, including [7] and [22], provides the technical capability to distribute document editing among multiple users. While these capabilities meet technical goals, the relevance to human performance is unclear. One system property of general interest is network delay, which designers often assume requires minimization to maximize usability. On the other hand, some usability limitations of otherwise effective groupware may yield to adaptations in work practice [18]. Some designers even suggest [25, 13] the benefit of delay warnings, so that users can adjust their strategies if they are aware of system conditions.

Not all groupware applications appear sensitive to delay. For example, Dourish and Bly [6] claim: *"We can tolerate a certain amount of delay; image updates may only occur every ten minutes, and so the user will not expect up-to-the-second information. However, delay must not be reflected in the manipulation of information in the interface, which must have good interactive response."* (p. 543) And later: *"Delays in source data, however, can be tolerated, which helps us achieve our goal of keeping network throughput low."* While we do not doubt the appropriateness of such assertions for the task in question, the underlying task theory and measurement methods are not sufficiently operationalized to permit generalization to other applications, making it difficult to evaluate either concern for delay or the efficacy of compensatory strategies.

1.1 Purpose

In this study we evaluate the performance consequences of simulated network delay in multi-synchronous collaborative document editing. Multi-synchronous collaboration maintains multiple, simultaneous streams of activity which continually diverge and synchronize [5]. In this setting, one user's changes appear to other users with some delay, creating potentially inconsistent perceptions of the document status.

The most directly related literature on the evaluation of collaborative editing tools is problematic. Grudin [11] attributes some of the challenge in evaluating any form of groupware to the time scale of realistic group activities, which can span weeks or months. Field and usability studies attempt to address this limitation, with extended periods of observation or retrospective questionnaires. Field studies [23] indicate the numerous challenges of developing collaborative writing tools. These include enhanced document quality, accessibility and reduced production time. Usability studies (e.g., [17, 1]), while informative of the relevant dimensions such as version control, do not provide quantitative behavioral evidence to guide the design of systems for collaborative use. Furthermore, Olson & Olson [18] offer an amusing illustration of the limitations of introspective methods for the evaluation of technology, and echo the human factors literature [14] that questions the relationship between subjective opinion and task outcome.

1.2 Related Quantitative Research

We highlight four implications from research related to the effect of delay in multi-synchronous activity: i) the need for an outcome metric, ii) the task time constant, iii) inherent task coupling and iv) the adoption of compensatory strategies to the limitations of collaboration technology.

Need for an Outcome Metric A quantitative relationship between groupware manipulations and outcome requires measurement of the latter. Olson [19] exemplifies the need for an outcome metric to evaluate the quality of the work produced with groupware. However, the measures can be overly domain specific. For example, although the study of collaborative editing tools

for instructional applications [3, 15] provides measures and methods, the analysis focuses on the evaluation of domain specific issues, e.g., instruction rather than the technology itself.

Task Time Constant A different literature captures the effect of delayed feedback on motor control tasks, for the individual user (e.g. [16]) and for collaborative motor control tasks characteristic of the gaming environment [13, 25]. These domains naturally provide an outcome metric. As a result, studies with game-like motor control tasks permit examination of the relationship between delay and performance. However, some of these studies identify performance decrements with delays as small 200 msec [13], on tasks with time constants (or turns) on the order of 700 ms. Furthermore, the results are mixed, demonstrating great tolerance for delay depending upon the metric (e.g., completion time versus errors). A persisting limitation in these studies is a coherent account of the relationship between properties such as the task time constant, delay and performance.

Task Coupling Generalization of the results from motor control studies to collaborative editing is unclear for reasons other than the task time constant and measurement issues. Coupling between sub-tasks, an issue that Olson & Olson [18] introduced, influences tolerance for delay. Olson & Olson mingle several potentially separable task properties in their discussion of coupling: the source of pre-requisites for task initiation, the agreement on task goals, the need for adaptation to local contingencies, and finally the resulting implications for communication.

Some sub-tasks require pre-conditions established under other subtasks. For example, one cannot edit grammar without an initial sentence to edit. Ill-structured problems [20] with poorly specified goal states exacerbate subtask coupling because the manner of subtask completion can interfere with implicit goals. For example, one participant's text for a procedure for the methods section of a scientific publication will conflict with another participant's effort to compose prose for a trade journal. Participants can agree on requisite relationships between sub-tasks and task goals, and still encounter local contingencies that require coordination. For example, a busy co-author may wish to know *when* a particular subtask is complete in order to initiate another, even if the former is conceptually unrelated to the later.

Compensatory Strategies The above task properties of task time constant and coupling lay the foundation for strategic adaptation. For example, in order to decouple the dependencies between activities, participants slow down [12] and partition coupled subtasks differently, which converts a coupled effort to the sum of individual efforts. Time-consuming communication is the backup for local uncertainties in the coordination of coupled tasks [24].

We conclude from the above research that while delay in the distribution of individual work is a potential influence on performance, the impact is a function of strategy and task. The quantitative impact of delay on task performance is unclear, particularly for tasks with larger time constants and discretionary coupling such as collaborative editing.

1.3 Experimental Editing Task

The present study addresses these limitations with a task time constant that lies between motor control and extended document preparation. A group of four participants i) located the release dates for an alphabetized list of movies and ii) re-sorted the list in chronological order.

The task has at least three methodological advantages. First, it supports a straightforward unidimensional outcome metric. Second, the sorting facet of the task bears some similarity to the manual control tasks associated with gaming. Finally, the task requires the highest degree

of interactivity and dependency that we could anticipate in document preparation, and should therefore bound the tolerance for delay in collaborative editing. We address three questions:

1. What is the effect of delay on task outcome and process?
2. What is the effect of strategy on task outcome and process?
3. How does strategy interact with delay to affect task outcome and process?

2 Methods

2.1 Participants

Eighty students affiliated with a European university participated in this experiment, in mixed gender groups of 4.

The participants ranged in age from 21 – 27. All participants used French in their daily activities, although they had sufficient working knowledge of English to comprehend the movie titles in the task stimuli. An electronic announcement solicited participation. One of the researchers organized interested participants into sets of 4 and scheduled the session. All participants received a 10 Euro gift certificate for their participation.

2.2 Apparatus

The experiment was conducted using four GNU Linux desktop computers in a classroom setting. Participants were separated by partitions and could not directly observe other team members while they worked, although typing activity was audible. The server running the Etherpad-lite application was hosted on an Amazon Elastic Compute Cloud (EC2) instance located in the US East (Northern Virginia) Region. Each desktop ran the Mozilla Firefox web browser executing the Etherpad-lite web client application. Etherpad-lite hosted the task stimuli and a Chat dialogue facility (see Figure 1). User operations appeared color-coded in both the text and chat. Etherpad-lite relies on a client-server architecture where each client/user edits a copy of the shared document. When a user performed a modification it was immediately displayed on the local copy of the document and then sent to the server. The server merged the change received from the user with other user changes and then transmitted the updates to the other users. When a user edited a sequence of characters, the first change on the character was immediately sent to the server, while the other changes were sent at once only upon reception of an acknowledgement from the server. With each change sent to the server, it created a new version of the document. Gstreamer software enabled the video recording of user activity. We also instrumented Etherpad-lite to register all user keyboard inputs on the client side and to introduce delays on the server-side. The editor window displayed 50 lines of text. Users editing above the field of view of a collaborator could cause the lines within the collaborators' view to "jump" inexplicably. Such a property is consistent with the inability to view an entire document as it undergoes modification from multiple team members.

2.3 Task & Stimuli

Participants conducted a 10 minute search and sorting task, starting with an alphabetized list of movies. Participants first used the internet to locate the release year for each movie and then sorted the list in chronological order. The list contained 74 movies, extending beyond the window size of the editor.

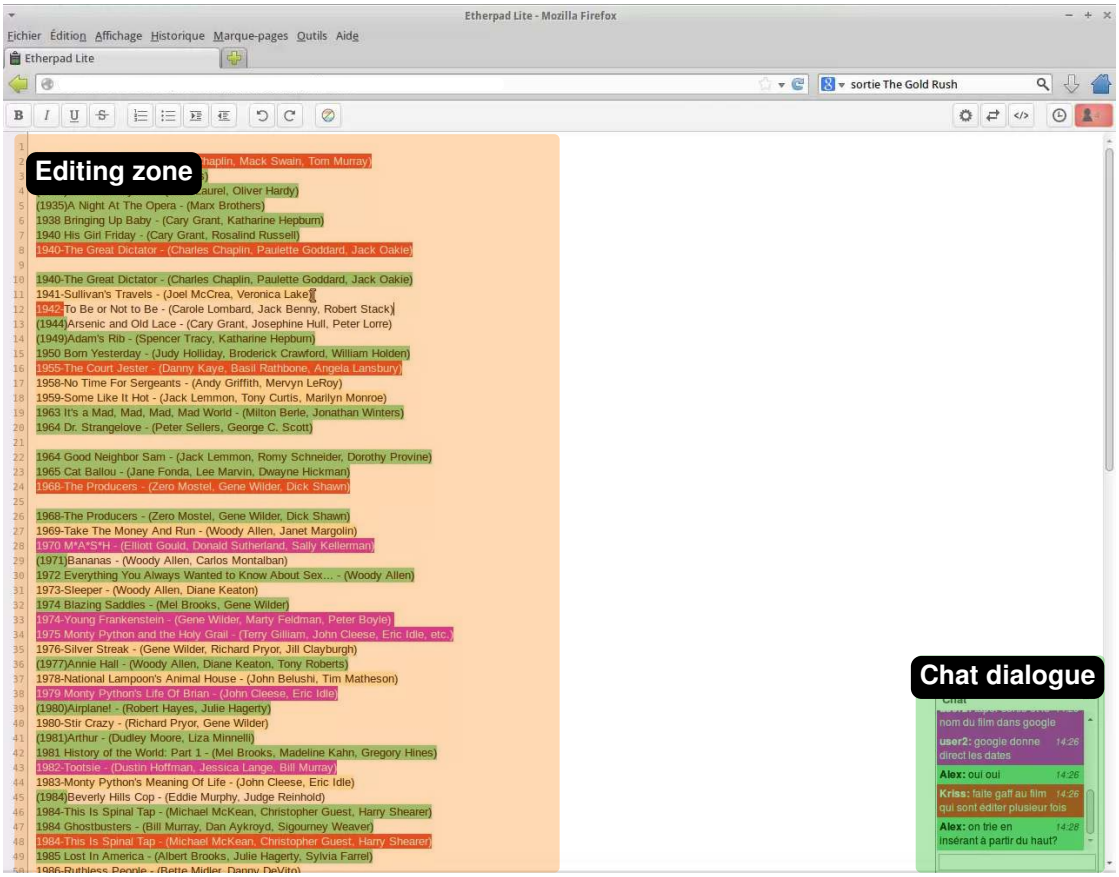


Figure 1: Etherpad-lite editor – each modification is highlighted with a color corresponding to the user who performed it.

2.4 Procedure

The entire procedure was approved by a US University IRB. Participants began the session with informed consent. The present sorting task was second in a three-task series. Scripted instructions (translated into English) for the sorting task follow: *"We will provide you with the list of movies. Your task is to search for the release dates of the movies and assemble a single list of movies sorted and labeled with their release dates. You can use the browser for finding the year of release of a movie. The year of release of a movie should be placed before the movie title and the movies should be sorted in an ascending order, starting from the oldest to the newest movie. You can work until we tell you to stop and you will have about 10 minutes for finishing the task. Please work as accurately as you can while still being efficient. You are free to coordinate your efforts with your teammates throughout the task using the chat interface at the bottom right side of the screen"*. The experimenters suspended the sorting task after approximately 10 minutes. Participants completed a third task (not reported here) prior to a debriefing survey.

2.5 Design

The sorting task was conducted with four teams of 4 participants for each level of the continuous independent variable Delay, tested at 0, 4, 6, 8 and 10 seconds in addition to the 100 msec delay inherent in the EC2. While participants viewed their own document changes in real-time, they viewed other participants' changes according to delay condition. Chat was implemented in real time for all conditions. Delay conditions were tested in random order, and all groups experienced a single level of delay across the three-task session.

2.6 Dependant Measures

We examined sorting accuracy as an outcome measure. We also examined a set of process measures including strategies, average time per entry, chat behavior, number of collisions in text editing and survey responses.

Sorting Accuracy is potentially sensitive to the scoring metric. We sought a metric that would reflect the accuracy of the list as a function of the movie dates and movie position in the list. Sedgewick [21] asserts insertion sort as the most likely strategy for human sorting, providing the justification for this metric here. Insertion sort iterates over an input list of elements and generates an output sorted list. At each iteration, an element in the input list is removed and inserted in the proper location within the sorted list, terminating when no more input elements remain. The insertion sort metric quantifies the distance between the input list and the output sorted list. Here the group provides the input list and the output list is the target list of movies, ordered according to their release dates. The distance between an element in the input list and the corresponding element in the sorted list is measured in terms of the number of swaps between adjacent elements required to place the input element properly in the sorted list. We normalized this distance with the distance in the worst case scenario, i.e. when the input list is sorted in reverse order. We additionally had to accommodate duplicated or missing movies, or movies with incorrect release dates. Therefore we eliminated the duplicated movies and the movies with an incorrect release date from the final list of movies generated by each group. We also eliminated from the output list the missing movies in the input list. The distance computed by the insertion sort metric was adjusted to be proportional to the number of movies that are not duplicated and for which users assigned the correct release dates. The formula that we used for each group score is provided below:

$$\left(1 - \frac{\#Swaps}{\#SwapsWorstCase}\right) \times \#Movies$$

$\#Swaps$ represents the total number of swaps between adjacent movies required using an insert sort method on the group's final list of movies. $\#SwapsWorstCase$ represents the total number of swaps between adjacent movies required by an insert method in the worst case, i.e. when the list of movies contains the movies in a descendant order according to the release dates. $\#Movies$ represents the number of movies in the final list of movies after a removal of duplicated movies or those with an incorrect release date. Two co-authors independently coded the insertion sort metric in different programming languages with identical results.

Average Time Per Entry was computed as the period of activity in question divided by the number of characters input. Because the task characteristics potentially changed over the 10 minutes, with the first half corresponding to the identification of movie dates and the second potentially corresponding to the sorting, we also calculated separate average response times for the first and second halves of the session.

Chat behavior was quantified as the number of turns, the number of words, agreement words (yes and OK), group oriented pronouns (You, your, one, us, who, each one, someone, no one,

others) and ego oriented pronouns (I, my, me, mine). We examined agreement words, group-oriented pronouns and ego-oriented pronouns as a function of the number of words.

Collisions was measured as the number of concurrent moves of the same movie by different users, resulting in movie duplication. This occurs when the action of one participant is invisible to another participant at the initiation of action. Etherpad-lite internally represents repositions as a delete-insert pair. Concurrent moves therefore result in multiple insertions, e.g., duplicates. A program for tallying collisions iterated over the revisions on the server, for each movie determining a history of operations that deleted or inserted that movie, the participant that performed that operation, and the operation timestamp. This identified duplicates and the time elapsed for any duplicate correction. We report on the total number of collisions, on the number of resolved collisions and on the time elapsed for resolving the collisions. We confirmed this account with the video recordings.

Strategies emerged through detailed analysis and are described below.

Survey responses examined here include:

Which exercise did you find most difficult? Why?

Quel exercice vous a semblé le plus difficile ? Pourquoi ?

Did anything annoy you about the text editor? If, yes, why?

Quelque chose vous a-t-il gêné dans l'éditeur de texte ? Si oui, quoi ?

What was the impact of the collaborative editing tool for sorting the list of films? Explain.

Quel a été l'impact de l'outil d'édition collaboratif pour le tri de la liste de films ? Expliquez.

3 Results

We provide results in three subsections, organized by measures. First we examine task strategies. Next we examine task outcome, followed by several measures of task process. For both outcome and process measures, we conduct regression modeling to describe our results, using Delay condition and Strategy as predictors, and follow up with simple effect analyses by Strategy. We examine additional facets of process in the indicators of coordination as apparent in Chat and survey data. We set our α at .1 to both compensate for low power in group level analyses [4, 10, 9] and in the case of the 1 df F-ratio, to mimic the one-tailed test corresponding to the expectation that delay will not benefit performance. We conclude the results with survey responses.

3.1 Strategies

As we had no a priori hypotheses about how users would divide up the work, we developed a coding scheme based on a review of the videos, supplemented by the chat discussion. Four strategies emerged:

- Sort at the end = sorting starts after all years have been added for all movies; there is no clear strategy for sorting the movies after the years were added
- Sort at the end by decade = after adding years, users sort movies into pre-established periods, i.e., is a more specific case of sort at the end.
- Continuous sorting = sorting is done immediately after adding a year for a movie
- Sorting distributed between participants = 1 or 2 users sort from the beginning of the task while the others add years

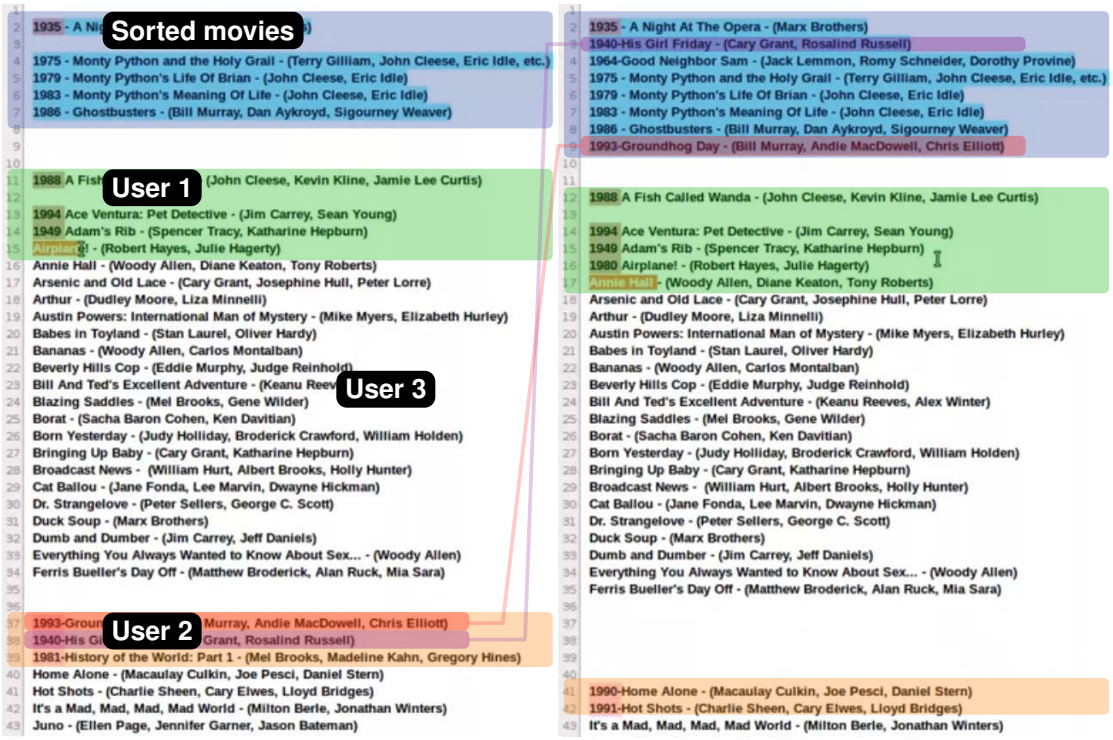


Figure 2: Continuous sorting strategy – jumping lines phenomenon.

We coded each video according to the dominant strategy in use, but consolidated the categories to avoid excessive partitioning of the data set. Henceforth, we label the first two strategies as "sort at the end" (Strategy 0) and the second two as "continuous sorting" (Strategy 1). "Sort at the end" enables loose coupling among participants at the beginning of the task, but leaves a highly coupled sorting task for end, with no pre-established assignments. "Continuous sorting" begins with a highly coupled distribution of work among participants. In Figure 2 we see two screen shots for the top of the document, separated by several seconds, for a group that adopted the continuous sorting strategy. User1 and User2 are adding years to several movies in the list, while User3 sorts the movies for which years were already inserted. User1 and User2 are experiencing a "jumping lines" phenomenon while adding years, (i.e., movement in the line position that occurs with an insertion earlier in the list). Here, while User1 adds the year for the movie "Airplane! - (Robert Hayes, Julie Hagerty)" located at line 15, the title of this movie jumps to line 16 due to User3's prior addition.

Although as a post-hoc variable Strategy was not balanced across conditions, there was no linear relationship between Strategy and Delay (adjusted $R^2 = -.05$, $p = .77$). However, because Strategy was not balanced, we routinely conducted simple effect analyses below to examine the effect of condition by separate levels of Strategy.

3.2 Outcome Measure

An insertion sort metric served as the outcome measure. Figure 3 displays the relationship between Delay, Strategy and insertion sort.

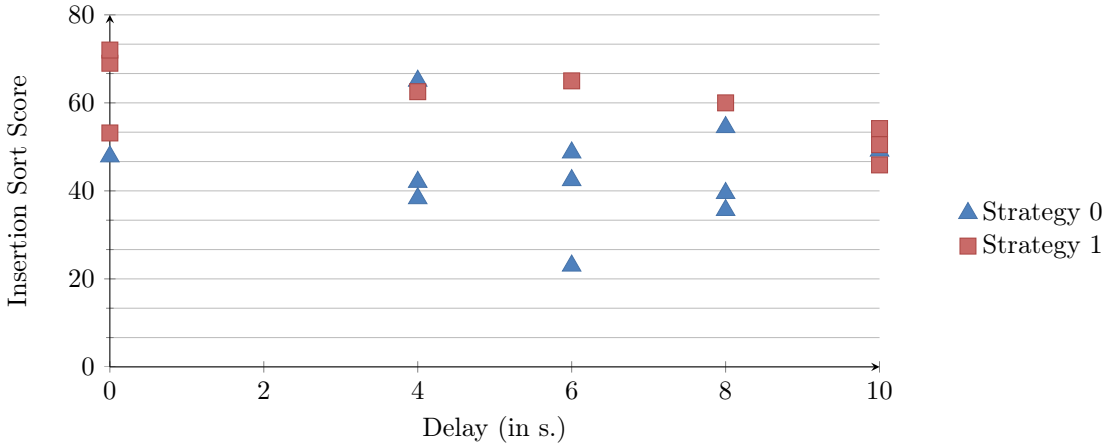


Figure 3: Insert Sort Score by Delay, separated by strategy. Strategy 0 groups pursued sorting after finding movie years. Strategy 1 groups pursued continuous sorting. Two overlapping data points for Strategy 1 in the 0 delay condition and two in the 10 second condition were artificially, slightly separated.

Strategy alone accounts for insertion sort score, $r(18) = .68$, adjusted $R^2 = .34$, $F(1, 18) = 10.89$, $p = .004$, $b_0 = 44.17$ $t(18) = 14.54$, $p < .001$, $b_1 = 4.53$, $t(18) = 3.30$, $p = .004$. A model with both Delay and Strategy just misses significance for Delay $r(18) = .58$, adjusted $R^2 = .24$, $F(2, 17) = 6.46$, $p = .011$, $_0 = 50.44$ $t(17) = 10.89$, $p < .001$, Delay $b_1^* = -.31$, $t(17) = -1.73$, $p = .102$, $r_{responsetime(delay,strategy)} = -.31$, Strategy $b_2^* = .59$, $t(17) = 3.35$, $p = .004$, $r_{responsetime(strategy,delay)} = .59$.

We examined the 9 groups who pursued Strategy 1 separately from the 11 groups who pursued Strategy 0. Among the 11 Strategy 0 groups, Delay does not predict insertion sort ($r(9) = .11$, adjusted $R^2 = -.10$, $F(1, 9) = .12$, $p = .742$). Among the 9 Strategy 1 groups, a linear model for Delay predicts insertion sort score ($r(7) = .69$, adjusted $R^2 = .39$, $F(1, 7) = 6.21$, $p = .042$). The intercept for the linear model is 66.46, $t(7) = 17.69$, $p = .000$ and the unstandardized slope is -1.38, $t(7) = -2.49$, $p = .042$. That is, each increment in Delay decrements the outcome measure by 1.38 Insertion Sort score units. A quadratic model does provide a better account for Strategy 1 groups ($r(7) = .75$, adjusted $R^2 = .50$, $F(1, 7) = 8.98$, $p = .020$). The intercept for the quadratic model is 65.99, $t(7) = 21.15$, $p = .000$ and the unstandardized Delay slope is $-.15$, $t(7) = -3.00$, $p = .020$. This model raises the possibility that Delay condition 10 results in qualitatively different behavior than the other conditions.

3.3 Process Measures

We also examined the average time between task inputs based on client recordings. Software error caused the loss of data for 4 groups. At the group level we used regression analysis to describe our results, treating Delay condition as a continuously valued independent variable. At the participant level, we used a nested ANOVA, using Delay condition as a categorical variable. The nested analysis allowed us to determine whether significant group effects precluded analysis of the Delay main effect. In general, we examined Delay condition and Strategy as independent

variables, with tests based on Type III Sums of Squares to account for the unbalanced design and an $\alpha = .10$ due to the small number of groups [4, 10, 9].

Significant group effects precluded statistical analysis of response time data across the entire experimental session. We proceeded with response time data from the session's first 5 minutes, where group effects were absent (see Table 1).

Condition	**Strategy 0**			**Strategy 1**		
	Mean	*SE*	*n*	Mean	*SE*	*n*
0	–	–	0	2323.68	57.83	3
4	2784.26	50.92	2	–	–	0
6	3916.95	606.91	3	2126.58	–	1
8	5352.10	470.48	2	3749.46	–	1
10	2584.99	–	1	3412.43	470.84	3

Table 1: First Half Session Response Times (msec) by Strategy.

Group response time over the first 5 minutes accounts for insertion sort score, ($r(14) = .49$, adjusted $R^2 = .19$, $F(1, 14) = 4.48$, $p = .053$, $b_0 = 70.37$ ($t(14) = 7.20$, $b_1 = -.01$, $t(14) = -2.12$, $p = .053$). Slowing down decreases outcome. However, due to the possibility of an alternative strategy in Delay condition 10, we plotted the relationship between insertion sort and response time for the groups in Delay condition 10 separately and note a positive relationship between group response time and insertion sort score (see Figure 4).

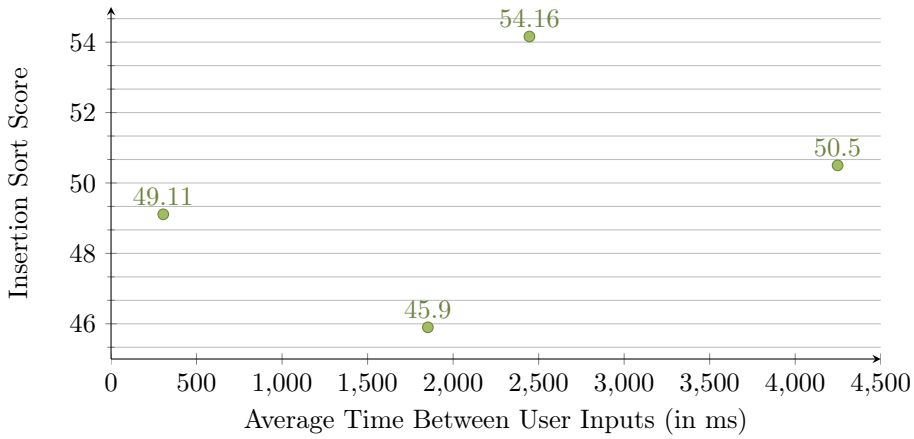


Figure 4: Positive relationship between group response time and insertion score for Delay condition 10.

Delay alone accounts for response time, $r(14) = .45$, adjusted $R^2 = .15$, $F(1, 14) = 3.56$, $p = .080$, $b_0 = 2504.77$ msec $t(14) = 4.81$, $p < .001$, $b_1 = 141.84$, $t(14) = 1.89$, $p = .080$. A model with both Delay and Strategy weakens the Delay effect, likely due to multicollinearity of the predictors $r(14) = .58$, adjusted $R^2 = .24$, $F(2, 13) = 3.34$, $p = .067$, $b_0 = 3014.57$ msec $t(13) = 5.18$, $p < .001$, Delay $b_1^* = .40$, $t(13) = 1.74$, $p = .105$, $r_{responsetime(delay,strategy)} = .39$, Strategy $b_2^* = -.37$, $t(13) = -1.64$, $p = .125$, $r_{responsetime(strategy,delay)} = -.37$.

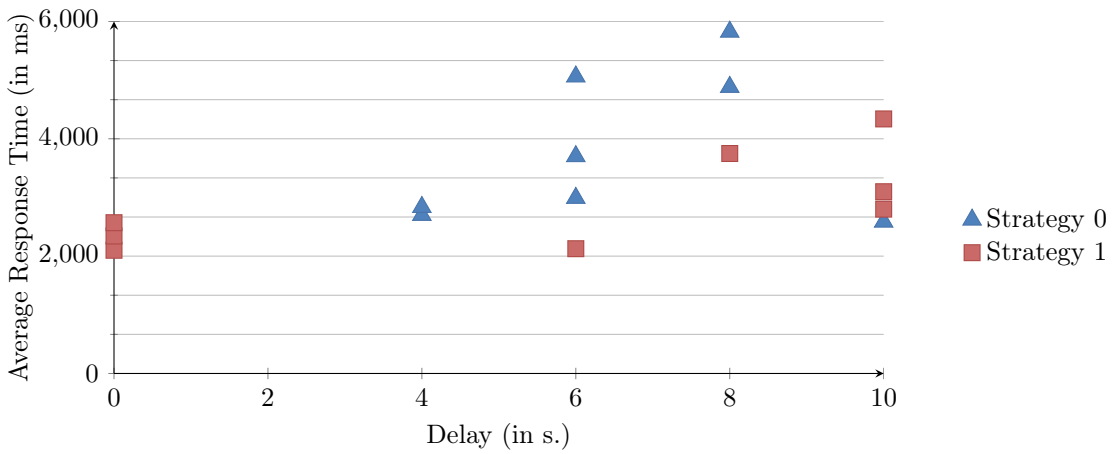


Figure 5: Average response time by Delay from the first half of session.

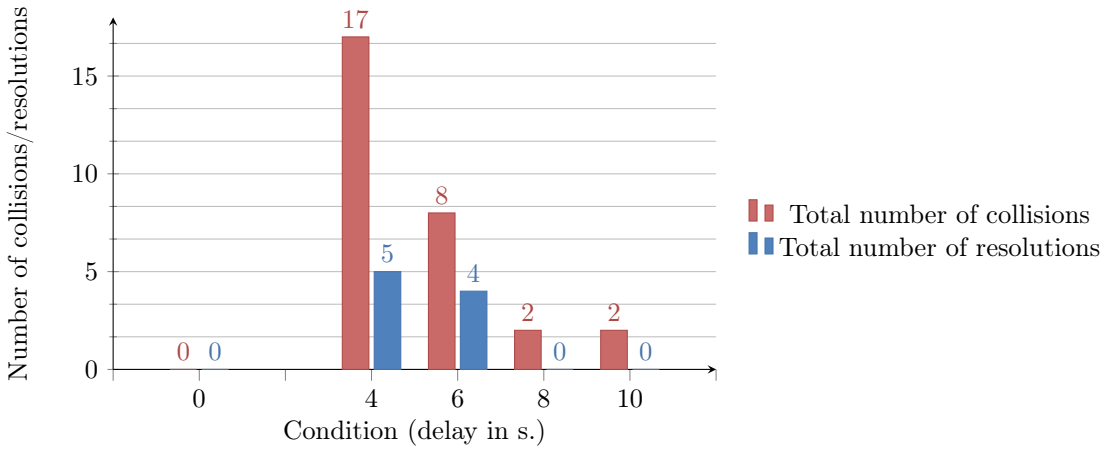
As suggested in Figure 5, separate Strategy models suggest quadratic effects of Delay on response times. For Strategy 0 the best model missed overall significance due to the intercept $r(6) = .75$, adjusted $R^2 = .38$, $F(2,5) = 3.12$, $p = .132$, $b_0 = 4103.58$ msec $t(5) = -1.54$, $p = .183$, Delay $b_1^* = 5.05$, $t(5) = 2.42$, $p = .060$, $r_{responsetime(delay,delay^2)} = .72$, Delay^2 $b_2^* = -4.81$, $t(5) = -2.30$, $p = .070$, $r_{responsetime(delay^2,delay)} = -.69$. For Strategy 1, the best model had an $r(6) = .71$, adjusted $R^2 = .42$, $F(1,6) = 6.12$, $p = .048$, $b_0 = 2281.23$, $t(6) = 7.02$, $p < .001$, Delay^2 $b_1^* = .71$, $t(6) = 2.47$, $p = .048$.

Collisions The management of sorting collisions appears to differ by strategy, as shown in Table 2 and Figure 6. The graph shows that Strategy 1 groups generally caught their collisions, and therefore preserved their insertion sort scores, until condition 10. In contrast, Strategy 0 groups did not catch their collisions, resulting in outcome score decrements.

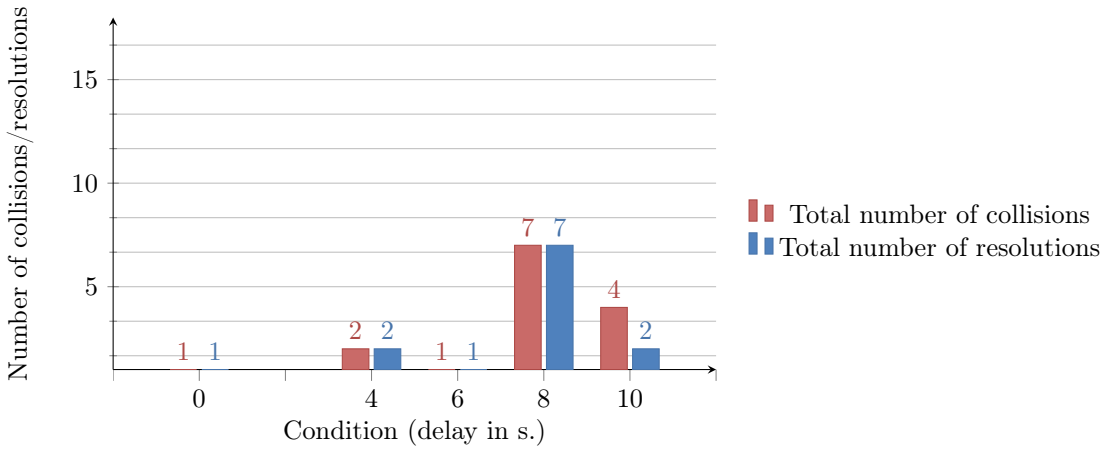
Condition	Strategy 0		
	$\#C$	$\#R$	Times of $\#R$
0	0	0	-
4	0	0	-
4	10	5	1′1″, 8″, 2′41″, 1′17″, 1′34″
4	7	0	-
6	3	3	8″, 1′, 26″
6	4	0	-
6	1	1	0″
8	2	0	-
8	0	0	-
8	0	0	-
10	2	0	-

Condition	Strategy 1		
	$\#C$	$\#R$	Times of $\#R$
0	0	0	-
0	1	1	0″
0	0	0	-
4	2	2	7″, 40″
6	1	1	2′12″
8	7	7	1′51″, 8″, 1′4″, 1′35″, 1′20″, 9″, 1′3″
10	2	2	20″, 21″
10	1	0	-
10	1	0	-

Table 2: Number of collisions ($\#C$) / resolutions ($\#R$) by Condition/Strategy.



(a) **Collisions with Strategy 0**



(b) **Collisions with Strategy 1**

Figure 6: Collision and resolution count by strategy.

Chat We examined chat metrics as predictors of insertion sort score. Proportion of accord words predicted insertion sort score ($r(18) = .54$, adjusted $R^2 = .25$, $F(1, 18) = 7.24$, $p = .015$, $b_0 = -40.00$, $t(18) = 8.49$, $b_1 = 3.83$, $p = .000$, $t(18) = 2.69$, $p = .015$). We also examined Strategy and Delay as predictors of chat metrics. Of these analyses, only total words was sensitive to the independent variables. For Strategy 0, Delay predicts total words ($r(9) = .64$, adjusted $R^2 = .34$, $F(1, 9) = 6.22$, $p = .034$, $b_0 = 29.08$, $t(9) = .79$, $p = .448$, $b_1 = 14.32$, $t(9) = 2.49$, $p = .034$). For Strategy 1, Delay does not predict total words ($r(7) = .34$, adjusted $R^2 = -.01$, $F(1, 7) = .92$, $p = .368$, $b_0 = 123.36$, $t(7) = .5.39$, $p = .001$, $b_1 = -3.23$, $p = .034$, $t(7) = -.96$, $p = .368$).

3.4 Survey Data

Which exercise did you find most difficult? Why? Those who found the film sorting task most difficult provided explanations indicated in Figure 3. Delay condition follows the explanation in question. All but one participant felt that success depended upon the sorting strategy.

Did anything annoy you about the text editor? If so, what? We identified and tallied two types of complaints. The first complaint concerned the perception of delay in the propagation of changes from one participant to another. The second complaint concerned movement of the task list in the editing window that occurred when users repositioned movies in the sorted list. Two experimenters coded the chat comments for these complaints independently and resolved the three cases of discrepancy among the 160 judgements.

Condition	**Lag Awareness**		**Jump Awareness**	
	Strategy 0	Strategy 1	Strategy 0	Strategy 1
0	.00(.00)	.09(.33)	.00(.00)	.18(.67)
4	.27(.33)	.00(.00)	.091(.33)	.25(1.00)
6	.00(.00)	.00(.00)	.333(.67)	.00(.00)
8	.25(.67)	.25(1.00)	.000(.00)	.00(.00)
10	.75(1.00)	.25(1.00)	.25(1.00)	.08(.33)

Table 4: Delay Awareness of Users (Groups).

Accounting for missing data and distribution of strategies, the maximum number of complaints is 35 for Strategy 1 and 43 for Strategy 0. Table 4 presents the proportions within a Delay condition who complained, for individual participants, and for the number of groups for which at least one user expressed a complaint. For example three-quarters of the participants in Delay condition 10, Strategy 0 complained of lag, constituting at least one member in every group. When aggregated to the group level, most or all groups complained of lag with 8 and 10 second delays in the propagation of user input. About half of the groups complained of list jumping. Unlike lag awareness, this does not appear to be related to Delay in a straightforward manner but rather a Strategy-Delay interaction. In particular, Strategy 1 (continuous sort) appears vulnerable to line jumping at low levels of Delay, presumably when users were able to work faster.

Interface Ratings Separate nested ANOVAs of the Likert scale ratings for the interface indicate an effect for strategy, $F(1, 18) = 12.44$, $p = .002$ (Strategy 0 $M = 6.86$, $SE = .23$, $n = 43$; Strategy 1 $M = 7.97$, $SE = .22$, $n = 35$).

4 Discussion

An artificial document editing task captures the upper limit of dependency and interactivity in collaborative editing, and permits the measurement of task outcome. Here we return to our original questions regarding the relationship between delay and strategy on process and outcome, before turning to methodological implications and future work.

Participant Explanations	**Delay**
You had to be synchronized to avoid looking for release dates on the same films. *Il fallait être synchronisé pour ne pas faire la recherche des dates de sortie des mêmes films* The work accomplished did not really take advantage of the platform used. *Le travail effectué n'a pas vraiment tiré partie des avantages de la plateforme proposée*	0
Sorting seemed to take lots of time, due to a lack of consensus over the method to use, which could go in any way. *Le tri semble prendre beaucoup de temps; faute de consensus sur la méthodologie à utiliser, ça peut aller dans tous les sens* We lacked time. Perhaps it would have been necessary to divide the task differently: two people to find the dates; two people to sort the list. *On a manqué de temps. Il aurait peut-être fallu répartir les tâches différemment: deux personnes pour trouver les dates; deux personnes pour trier la liste* We found the dates fast enough but the sorting was not done efficiently. *Nous avons trouvé toutes les dates assez vite mais le tri ne s'est pas fait efficacement* We must be fast and well coordinated amongst ourselves to get through to the end. We should distribute the tasks from the beginning. *Il faut être rapide et bien coordonnés entre nous pour arriver jusqu'au bout. Il faut bien se répartir les différentes tâches dès le début* The problem is to coordinate the sorting of films well once the dates are found, there are probably doubles in the list moreover. *Le problème est de bien se coordonner pour le tri des films une fois les dates trouvées, il y a probablement des doublons dans la liste d'ailleurs*	4
Duplicate elements makes it difficult to classify. Because everyone did his own part at the same time, the lines come and go all the time and that can affect how one makes selections with the mouse. In the same way, the possibility of intervening at the same time allows for the titles to be recopied several times and therefore, to appear several times in the final list. *La multiplication d'éléments (texte) présents à l'écran rend le classement difficile. Comme tout le monde traite sa partie en même temps des lignes arrivent et partent tout le temps et cela peut avoir une incidence sur les sélections que l'on fait à la souris. De même, la possibilité d'intervenir en même temps implique que des titres de films soient recopiés plusieurs fois et donc, apparaissent plusieurs fois dans la liste définitive*	6
The utilized sorting algorithm was not very efficient. In our case, collaborative sorting consisted of merge sorting. *Algorithme de tri utilisé, pas vraiment efficace. faire le tri en collaboratif dans notre cas consistait à utiliser un tri fusion*	8
Slow start, we had a good technique but were too slow, impossible to finish on time. *Départ lent, on a pris une bonne technique mais on a été trop lents du coup impossible de finir le tri à temps*	10

Table 3: Explanations for the Difficulty of the Movie Sorting Task.

4.1 1) What is the effect of delay on task process and outcome?

Because of the interaction of delay with strategy, we have just a few comments about the general effect of delay. First, the general effect of delay is to slow down performance, which in this case hinders task outcome. Second, we suggest that the effect delay is a function of the task time constant. In our case, with a task time constant on the order of 3.5 seconds, the effective manipulations are much larger than the 700 msec characteristic of motor control tasks, with problems becoming apparent at 8 and 10 seconds. Third, the effect of a system property such as delay can be small relative to strategies. Fourth, the effect of delay need not be linear, and not simply because of weak effects at low levels of disturbance.

Complaint about delay was a low frequency event at the level of individual participants. This makes sense because participants would have had to experience the direct consequence of delay in order to detect it. Collisions are one such consequence, which depending upon strategy, participants did not always recognize and repair. Nevertheless, at the group level, most groups had at least one participant complain about delay at the highest levels.

4.2 2) What is the effect of strategy on task process and outcome?

As with the main effect of delay, we postpone much of our discussion of the strategy effect to a later section, where we can address the interaction between strategy, delay and performance. First, we note here that the relationship between speed and insertion sort scores, is not *in general* a speed accuracy tradeoff. In general, slowing down does not improve scores, because slowing down results in fewer properly dated, positioned movies which will decrement the insertion sort score. Second, a strategy effect appears in the outcome measure, response time, and interface ratings. Continuous sort groups liked the interface better than sort at the end groups. This could reflect satisfaction with the higher level of task success. Finally, we noted above the relative effect size of strategy with respect to delay. In fact, delay effects often emerge as significant only with simple effect analyses, although strategy survives as an effect on its own.

4.3 3) How does strategy interact with delay to affect task process and outcome?

The effect of delay depends on strategy. Such an interaction between strategy and experimental manipulation on outcome is consistent with prior studies in game-like motor control environments. However, somewhat counterintuitively, and unlike previous research, the overall superior strategy does not overcome the effect of delay. In fact, the insertion sort score declined with delay for continuous sort, but did not for sort-at-the-end. We suspect that continuous sort entails more coupling, because years must be in place prior to positioning, and because text position is changing frequently throughout the entire task as sorting proceeds. Even continuous-sort participants in the 0 and 4 second delay conditions complained about the "jumping" line positions. To manage the coupling in continuous sort, we see participants slow down with delay. However, the negative slope on the insertion sort metric for continuous sort relative to the flat slope for sort-at-the-end suggests that the continuous sort strategy is only adaptive within a range of delay. Untested levels of delay could actually result in worse performance for continuous sorting than a sort-at-the-end strategy.

The sort-at-the-end strategy did not encounter coupling until the later phases of task completion. However, the chat metric suggests that sort-at-the-end requires more local coordination as delay increases. Thus the coordination established by formal agreement at the outset in continuous-sort appears to favor efficient communication over the ability to respond to local per-

turbations. On the other hand, sort-at-the-end appears to favor the ability to respond to local perturbations at the expense of efficient communication.

A different indication of the interaction between strategy and delay appears in the groups who experienced 10 seconds of delay. We noted a surprising relative reduction in response time as well as insertion sort score with 10 seconds of delay. Unlike the reverse speed-accuracy tradeoff apparent in the overall data set, a graph of the relationship between insertion-sort score and average response time for the four groups at delay 10 suggests a speed-accuracy tradeoff. This in turn suggests a different type of strategy. Our interpretation of this finding is an overall reduction in the precision of movie placement with rapid movement, resulting in more frequent changes to approximate the final location of the movie in question.

4.4 Methodological Implications

Our methodological point transcends any limitations of power and an unbalanced design. The evaluation of collaborative editing software requires outcome and process metrics, examined in a controlled setting to support causal assessment, and a statistical analysis that includes both significance testing and percent-variance-accounted-for. In the absence of this methodological rigor we cannot attribute performance changes to fine-grained quantitative system properties, and we cannot assess the relative importance of technical effort to reduce purportedly problematic system properties. Nevertheless, we acknowledge several persisting challenges in the pursuit of this goal.

The average response time metric used here is subject to criticism concerning the units of analysis tallied per unit time. While this issue is typically approached with respect to artifacts introduced by the servers that count complex operations as one, the proper tally is not at all obvious. Our solution was to weight the operation by number of characters manipulated, but number of words might be plausibly used instead. This thorny issue in the unit of analysis pervades all efforts to model workload functions and operator throughput at higher levels of task complexity, and we do not claim to have settled it here.

We exploited a task-specific outcome metric, insertion-sort, which is subject to criticism on grounds of generalizability to the problem of collaborative editing. Position accuracy is hardly a standard metric for prose quality. Moreover, the observed relationship between outcome and process is atypical. Unlike the task examined here, slowing down typically enhances task accuracy [26]. Certainly other metrics for prose quality, such as the Flesch grade level [8] and more recently, Natural Language Processing techniques that assess text coherence [2] merit exploration in the evaluation of collaborative editing. However, we have several concerns with reliance on such metrics and corresponding paradigms. Chief among these is the longer task time constant, which will extend the duration of observation and likely increase measurement noise. Individual differences and group composition further complicate reliance on prose quality metrics. Perhaps most important, the longer task time constant likely implies greater tolerance for delay, well within the capabilities of contemporary technical capability to distribute document edits among multiple users. Thus, pursuing such inquiry may be an important academic exercise in establishing the quantitative function that relates delay to performance, but provides little motivation for engineering efforts to reduce distribution delay.

4.5 Future work

Performance on a family of related tasks will help to address the relationship between delay and task properties. We have data for the effect of delay on two other artificial editing tasks that vary both the task time constant and the degree of subtask coupling. The analyses presented

here suggest the need to add Delay levels, with 2 and 12 second delays and beyond. This will help determine whether the models that relate performance to delay are appropriately linear, or quadratic, with more rapid declines in task performance with delay. In particular the 2 second data point will clarify the need for combined linear and quadratic influences. Delays of 12 seconds and more will help to confirm the quadratic influence, and more important, test the implicit claim that the efficient, tightly coupled strategy can decrement scoring below the more loosely coupled strategy.

5 Conclusions

The general effect of delay on an artificial document editing task is to slow the individual participant, which for the present task, decrements the outcome metric. However, consistent with the literature with game-like tasks (e.g., [13, 25]), the effect of delay on document editing, as measured by outcome, depends on strategy. A tightly coupled subtask decomposition that enhances outcome in the presence of minimal delay becomes detrimental at higher levels of delay, potentially less effective than a more loosely coupled task decomposition at the beginning of the task. Nevertheless, a loosely coupled strategy at the beginning of the task leaves a poorly coordinated, tightly coupled sorting task to the end of the task, increasing the need for communication and hampering overall performance. Given the time constant of the present task, strategy is at least as important as delay in the distribution of participant inputs to the team.

6 Acknowledgments

The authors are grateful for financial support of the USCoast Inria associated team and a graduate assistantship funded by Inria. The fourth author is grateful for sabbatical support from the Department of Psychology, College of Science and Mathematics, Wright State University, Dayton OH. This work is partially funded by the french national research program STREAMS (ANR-10-SEGI-010).

References

[1] Andy Alder, John C. Nash, and Sylvie Noël. Evaluating and implementing a collaborative office document system. *Interacting with Computers*, 18(4):665–682, July 2006.

[2] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, March 2008.

[3] Coye Cheshire and Judd Antin. The social psychological effects of feedback on the production of internet information pools. *Journal of Computer-Mediated Communication*, 13(3):705–727, April 2008.

[4] Nancy J. Cooke, Jamie C. Gorman, Jasmine L. Duran, and Amanda R. Taylor. Team cognition in experienced command-and-control teams. *Journal of Experimental Psychology: Applied*, 13(3):146–157, September 2007.

[5] Paul Dourish. The parting of the ways: divergence, data management and collaborative work. In *Proceedings of the fourth conference on European Conference on Computer-Supported Cooperative Work*, ECSCW'95, pages 215–230, Stockholm, Sweden, 1995. Kluwer Academic Publishers.

[6] Paul Dourish and Sara Bly. Portholes: supporting awareness in a distributed work group. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 541–547, Monterey, California, USA, 1992. ACM.

[7] Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. Groupware: Some Issues and Experiences. *Communications of ACM*, 34(1):39–58, January 1991.

[8] Rudolph Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233, June 1948.

[9] Jamie C. Gorman and Nancy J. Cooke. Changes in team cognition after a retention interval: The benefits of mixing it up. *Journal of Experimental Psychology: Applied*, 17(4):303–319, December 2011.

[10] Jamie C. Gorman, Nancy J. Cooke, Harry K. Pedersen, Jennifer Winner, Dee Andrews, and Polemnia G. Amazeen. Changes in team composition after a break: Building adapative command-and-control teams. *Proceedings of the the Human Factors and Ergonomics Society Annual Meeting*, 50(3):487–491, October 2006.

[11] Jonathan Grudin. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM*, 37(1):92–105, January 1994.

[12] Carl Gutwin. The effects of network delays on group work in real-time groupware. In *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, ECSCW'01, pages 299–318, Bonn, Germany, 2001. Kluwer Academic Publishers.

[13] Carl Gutwin, Steve Benford, Jeff Dyck, Mike Fraser, Ivan Vaghi, and Chris Greenhalgh. Revealing delay in collaborative environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 503–510, Vienna, Austria, 2004. ACM.

[14] Sandra G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, 50(9):904–908, October 2006.

[15] Nobel Khandaker and Leen-Kiat Soh. Improving group selection and assessment in an asynchronous collaborative writing application. *International Journal of Artificial Intelligence in Education*, 20(3):231–268, August 2010.

[16] Mark Mulder, Marinus M. van Paassan, John .M. Flach, and Richard .J. Jagacinski. *Fundamentals of manual control*. Taylor & Francis, Boca Raton, FL, 2006.

[17] Sylvie Noël and Jean-Marc Robert. Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work*, 13(1):63–89, 2004.

[18] Gary M. Olson and Judith S. Olson. Distance matters. *Human-Computer Interaction*, 15(2):139–178, September 2000.

[19] Judith S. Olson, Gary M. Olson, Marianne Storrøsten, and Mark Carter. Groupwork close up: A comparison of the group design process with and without a simple group editor. *ACM Transactions on Information Systems*, 11(4):321–348, October 1993.

[20] Walter R. Reitman. Cognition and thought. an information processing approach. *Psychology in the Schools*, 3(2):189, April 1966.

[21] Robert Sedgewick. *Algorithms*. Addison-Wesley, 1983.

[22] Chengzheng Sun, Xiaohua Jia, Yanchun Zhang, Yun Yang, and David Chen. Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems. *ACM Transactions on Computer-Human Interaction*, 5(1):63–108, March 1998.

[23] Susan G. Tammaro, Jane N. Mosier, Nancy C. Goodwin, and Gabriel Spitz. Collaborative Writing Is Hard to Support: A Field Study of Collaborative Writing. *Computer-Supported Cooperative Work*, 6(1):19–51, 1997.

[24] Sébastien Tremblay, François Vachon, Daniel Lafond, and Chelsea Kramer. Dealing with task interruptions in complex dynamic environments: Are two heads better than one? *Human Factors*, 54(1):70–83, February 2012.

[25] Ivan Vaghi, Chris Greenhalgh, and Steve Benford. Coping with inconsistency due to network delays in collaborative virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '99, pages 42–49. ACM, 1999.

[26] Wayne A. Wickelgren. Speed-accuracy tradeoff and information processing dynamics. *Acta Psychologica*, 41(1):67–85, February 1977.

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399