# Attention-Based Management of Information Flows in Synchronous Electronic Brainstorming

Antonio Ferreira[1], Valeria Herskovic[2], and Pedro Antunes[1]

[1] Department of Informatics, University of Lisbon, Portugal
asfe@di.fc.ul.pt, paa@di.fc.ul.pt
[2] Department of Computer Science, Universidad de Chile, Chile
vherskov@dcc.uchile.cl

**Abstract** In this paper we argue for buffering group awareness information to mitigate information overload and help users keep up with the group. We propose an attentive groupware device, called the opportunity seeker, that leverages the natural alternation between a user doing individual work and attending to the group to automatically manage the timing and quantity of information to be delivered based upon each user's state of attention. We explain how this device can be applied to synchronous electronic brainstorming and present results from a laboratory experiment, which indicate that groups produced 9.6% more ideas when compared to the immediate broadcast of ideas. In addition, a user-level post-hoc analysis suggests that information overload was attenuated with the opportunity seeker as users had 7.5 seconds of extra uninterrupted time to think about and type an idea, which they began to write 6.4 seconds sooner, and completed in 4.2 seconds less time.

## 1 Introduction

Attention management is an important topic in our information-rich world and is gaining momentum in the Human-Computer Interaction (HCI) field as evidenced by recent research on Attentive User Interfaces (AUI) [1,2]. The main motivation for AUI is the recognition that as the needs for information and communication rise so do the costs of not paying attention and being interrupted. So, instead of assuming the user is always focused on the entire computer screen, AUI negotiate the users' attention by establishing priorities for presenting information.

Most research on AUI is directed towards single-user work and assumes user performance degrades with the number of simultaneous requests for attention. Therefore, researchers are enhancing input/output devices so that the user remains focused on a primary task without getting too much distracted by secondary—typically unrelated and unexpected—tasks, e.g., by using eye-gaze and body orientation sensors [3], statistical models of interruptibility [4], and displays capable of showing information at various levels of detail [5].

Regarding multi-user work, the research is situated in video conferencing [6,7], making the study of AUI for groupware systems a largely unexplored area. We present three arguments to promote further investigations on this matter.

Ferreira, A., V. Herskovic and P. Antunes (2008) Attention-Based Management of Information Flows in Synchronous Electronic Brainstorming. Groupware: Design, Implementation, and Use. 14th International Workshop, CRIWG 2008, Omaha, US, September 2008 Proceedings. R. Briggs, P. Antunes, G. Vreede and A. Read, vol. 5411, pp. 1-16. Heidelberg, Springer-Verlag.

Firstly, the convergence of AUI and groupware systems poses new challenges to researchers due to differences in individual and group work: a) people working in a group are more occupied with requests for attention because they have to manage more information flows; b) instead of doing a single extensive task, group members usually execute a series of intertwined tasks; c) group members have to explicitly manage the trade-offs of attending to the group and doing individual work; and d) in group work the primary and secondary tasks are typically related and may both contribute to the shared goal.

Secondly, the current emphasis of AUI applied to groupware systems is still, to the best of our knowledge, on evaluating the enhanced devices *per se* (for example, the perception of movement or sudden brightness changes [6]), in contrast with determining the outcomes of using these devices in work settings.

Thirdly, groupware researchers are designing systems that provide ever greater awareness information about the presence and actions performed by users on a group using devices such as radar views, multi-user scrollbars, and telepointers [8]. However, a problem with this trend is that it fails to recognise that sometimes more is less due to the limitations of the human attentive capacity.

Given this situation, we must consider the *group attention problem*: as the needs for collaboration rise so do the costs of not attending to the group and becoming overloaded with information.

We argue that this problem is poorly addressed by existing group awareness devices due to the lack of assumptions regarding human attention and because these devices require manual control of the type and quantity of information to be displayed, e.g., via filters, thus penalising individual performance.

This trade-off between limiting group awareness information and manual intervention by the users sets the stage for introducing an attentive device that automatically adjusts the delivery of group awareness information using a buffering technique grounded on each user's predicted state of attention. We explain how the device can address information overload in synchronous electronic brainstorming sessions and report the results of a laboratory experiment to evaluate group performance with and without the attentive device. Next, we discuss the validity of the model of user behaviour that we used for the brainstorming context and identify some limitations of this study. We conclude the paper with a summary of contributions and paths for future work.

## 2 Related work

The study of AUI for groupware systems is, for the most part, an unexplored research area, with the exception of video conferencing. GAZE-2 is a system developed to facilitate the detection of who is talking to whom in remote meetings [6]. It shows video images of the users' faces on the computer screen, which can be automatically rotated by intervention of eye-trackers placed in front of each user, so that the faces appear to be staring at the user who is speaking. In this way, group turn taking may be more natural and require fewer interruptions to determine who will speak next.

Another feature of GAZE-2 is the automatic filtering of voices when multiple conversations are being held at the same time. Depending upon the user in focus, the respective audio stream is amplified, and the other streams are attenuated. If the focus of interest suddenly changes, as sensed by the eye-tracker, the audio is again adjusted. To save network bandwidth, filters are also applied to the video images by decreasing their quality as the angle of rotation increases.

The eyeView system explores the GAZE-2 ideas in the context of large meetings. It controls the size of video windows, arranged side-by-side, as well as the users' voice volumes as a function of the user's current focus of attention [7].

GAZE-2 and eyeView utilise audio and video filters to manipulate the amount of group awareness information that users are exposed to during electronic meetings. However, we found no evidence that group work benefited. Instead, the literature describes technological evaluations via user questionnaires concerning the self-subjective perception of eye-contact and distraction, as well as changes in colour and brightness during camera shifts [6].

Some studies do address the evaluation of AUI from the perspective of task execution, but are restricted to single-user activity. One study measured the effects of interruptions on task completion time, error rate, annoyance, and anxiety, and suggests that AUI should defer the presentation of peripheral information until task boundaries are reached [9]. In another study, the effectiveness and efficiency of users were evaluated as they performed two types of tasks under the exposure of four methods for coordinating interruption, and the authors recommend that AUI should let users manually negotiate their own state of availability, except when response time for handling the interruptions is critical [10].

However, as we mentioned earlier, there are numerous differences in individual and group work, which opens an opportunity for doing research on AUI for groupware systems.

## 3 Addressing the group attention problem

To deal with the group attention problem—highlighting the need to keep users mindful of the group and mitigate information overload—we developed an attentive device for synchronous groupware systems, called the opportunity seeker, which collects group awareness information in a buffer and automatically manages the timing and quantity of information to be delivered to each user based upon his or her state of attention.

There is a trade-off in managing the delivery timing and quantity of group awareness information, in that too few updates may give the wrong impression about what the group is doing, while too many may provide up-to-date awareness information but be too distracting. We address this trade-off by leveraging the typical alternation between primary and secondary tasks in group work to find natural opportunities to interrupt the user. According to Bailey and Konstan [9] these opportunities should occur at the boundaries between consecutive tasks, i.e., for group work, at the transitions between the user doing individual work and paying attention to the group (see Fig. 1).

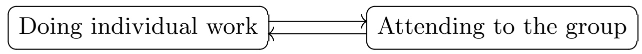


Figure 1: Natural task switching during group work.

Thus, regarding the delivery timing, the opportunity seeker only displays group awareness information to the user when s/he is likely *not* doing individual work. Concerning the limit on the quantity of information to deliver at once, the purpose is to avoid overloading the user if his or her work pace differs too greatly from the rhythm of the group.

### 3.1 Tackling information overload in electronic brainstorming

The rules of brainstorming [11] encourage users to do two cognitive tasks: the first is to produce as many ideas as possible because quantity is wanted; and the second is to read, or at least look at, the other users' ideas because combination and improvement of ideas is sought (cf. tasks in Fig. 1). In electronic brainstorming users can submit ideas in parallel, which puts more effort in the second cognitive task. As the number of ideas increases, e.g., because the group is inspired or group size is large, users may no longer be able to process the ideas, and may even become distracted by them, thus causing information overload.

It was for this work context that we created the first implementation of the opportunity seeker. The result is ABTool, or Attentive Brainstorming Tool, a custom-made tool for synchronous electronic brainstorming with built-in sensors of user performance that automatically manages the timing and quantity of ideas to be delivered to each user over a brainstorming session.

Two major challenges in applying the opportunity seeker to ABTool were to characterise how users work in a scenario with immediate broadcast of ideas to the group, and to detect task switching during electronic brainstorming activity. To this end we asked groups of five volunteers to simulate a distributed work setting by only using the tool to communicate, i.e., no face-to-face interaction was allowed. We recorded three types of events: a) user key presses while typing ideas; b) the moments when the user submitted an idea to the group; and c) the instants when group ideas were delivered to the user's computer screen.

Figure 2 shows a sample of the data we obtained and illustrates the results for an entire fifteen minute session, in which 152 ideas were produced.

From the evidence we collected three patterns of user activity emerged: a) users usually did not stop typing when they received ideas from the other users, thus, we assume they continued focused on the individual task of generating ideas; b) users typically paused after putting forward an idea, presumably to keep up with the group; and c) there were numerous periods of time with no typing activity (not shown in Fig. 2).

Based upon these three patterns, we hypothesise that a task boundary, i.e., an opportunity to display ideas from others, occurs when the user submits an idea to the group. In addition, new ideas should be delivered after a period of

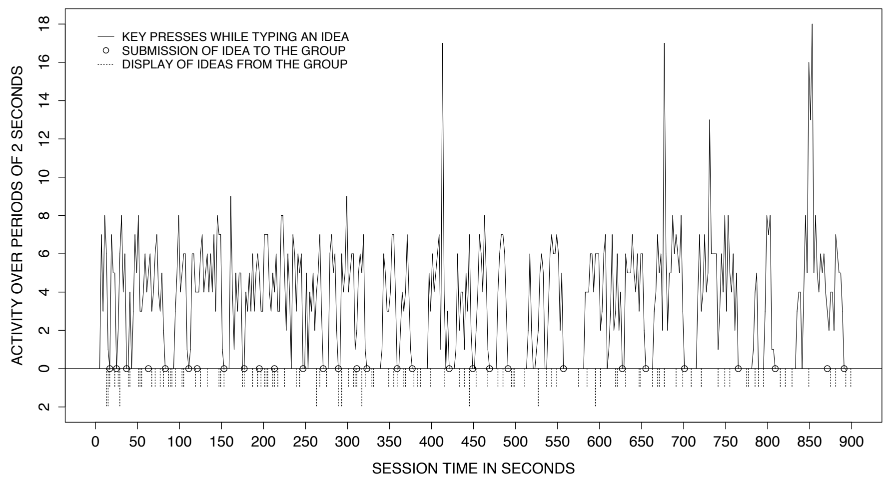


Figure 2: User and group activity during a brainstorming session with ABTool, with immediate broadcast of ideas to everyone on the group (i.e., with the opportunity seeker disabled). Above the X-axis are aggregated counts of user key presses. The spikes occurred when the user pressed the delete or cursor keys. The circles on the X-axis show when the user submitted the idea s/he was typing to the group. Below the X-axis are the instants in time when the user received ideas from the other users.

inactivity (currently, ten seconds) so that the user does not get the impression that the group is not producing ideas too.

Figure 3 shows the state transition diagram that models the behaviour of the user as assumed by the opportunity seeker on ABTool (also cf. Fig. 1).

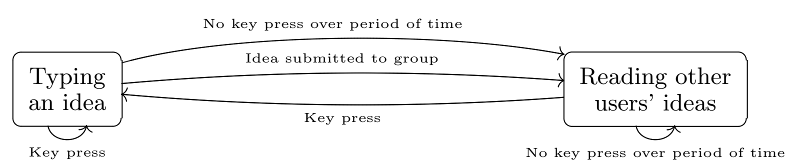


Figure 3: Model of user behaviour assumed by the opportunity seeker on ABTool.

Another feature of the opportunity seeker is that it imposes a limit on the number of ideas from others that can be displayed at once (currently, ten). This is to avoid overloading the user, e.g., by filling up the entire computer screen with new ideas, when the user is working at a slower pace than the other group members. Figure 4 shows a simulation that exemplifies the delivery of ideas with the opportunity seeker compared to the immediate broadcast of ideas.

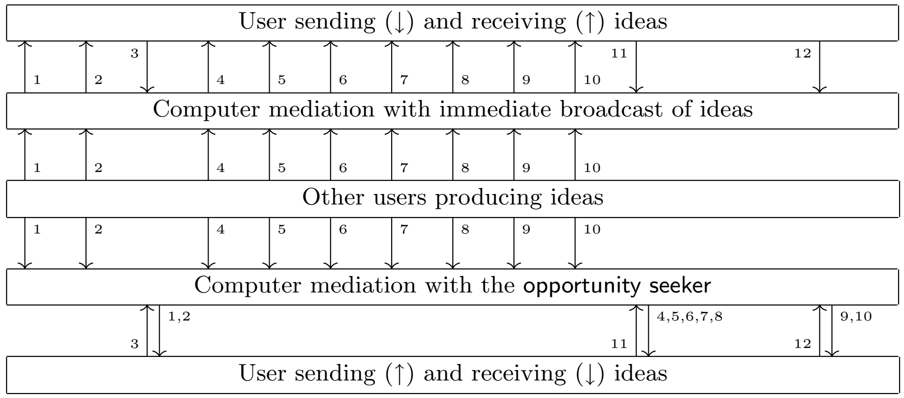


Figure 4: Simulation of group and user activity during a brainstorming session with immediate broadcast of ideas (*upper region*) and with the opportunity seeker (*lower region*). In both cases the user produces three ideas (numbered 3, 11, and 12) but the exposure to the nine ideas s/he received from the other users is different. For illustration purposes, we do not show the propagation of ideas 3, 11, and 12 to the group, and limit the number of ideas delivered at once to five.

### 3.2 Software architecture and design

Technically, ABTool is characterised by a client-server architecture, in which the server mediates the group information flows. The server also collects performance data, which are stored in an XML log. The purpose of the clients, one per user, is to receive input from the users and pass it on to the server, and to display new ideas as they become available from the server.

ABTool is written in C# and is based upon the Microsoft .NET Framework 2.0. Communication between the clients and the server is done via TCP/IP sockets and all messages (ideas, key presses, users joining or retiring the group, sessions starting or ending) are automatically serialised and deserialised using BinaryFormatter objects attached to NetworkStream instances.

Within the client and server applications, messages are propagated using events, to which consumer objects can subscribe themselves. Given that almost all classes on ABTool handle message events, namely the user interfaces, the opportunity seeker, and the classes responsible for receiving and sending messages from/to the network, we defined an IHandlesMessages interface and a default implementation for it, DefaultHandlesMessages, which relies on reflection to allow those classes to delegate the determination of the method to run as a function of the type of message associated with the event.

Figure 5 shows that the opportunity seeker derives from the AttentiveDevice generalisation, which actually implements immediate delivery of ideas from the users to the group. The OpportunitySeeker class alters this default behaviour by maintaining separate buffers, one per user, containing ideas that have been put forward by the other users. The buffer is stored in the UserNode, which also keeps a Timer object that every verificationPeriod milliseconds verifies the time of the most recent key press by the user, and if it was more than activationTimeSpan milliseconds ago, then it delivers up to ideasAtOnce ideas to the user.

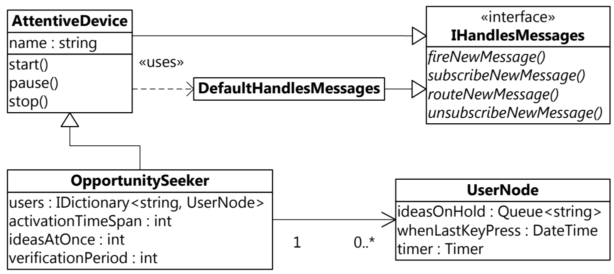


Figure 5: Class diagram showing details of the opportunity seeker on ABTool.

The AttentiveDevice and OpportunitySeeker classes implement three methods: start() is run when a session starts or resumes; pause() is executed when, for some reason, the session needs to be paused; and stop() is run at the end of a session. Other methods handle the reception and forwarding of messages, but we omitted those for brevity.

To conclude the presentation of ABTool, we show in Fig. 6 two screen shots of the client application with the opportunity seeker running.

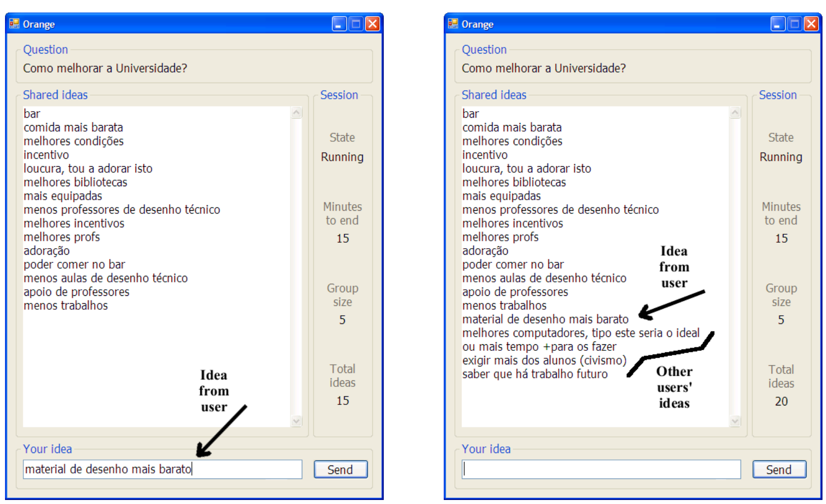


Figure 6: Opportunity seeker managing the delivery of ideas on ABTool. *Left:* while typing an idea, the user receives no new ideas from the group. *Right:* when the user submits an idea to the group, new ideas from others are displayed.

# 4 Laboratory experiment

We now describe a laboratory experiment that we set up using ABTool to test the hypothesis that group performance, measured by the number of ideas produced, improves when groups are exposed to the opportunity seeker device.

## 4.1 Participants

A total of 11 groups of 5 people, for a total of 55 volunteers (44 men and 11 women) participated in the experiment. The median age was 23 years (min. 20 and max. 29). 51 participants were students (40 undergraduate, 10 MSc, 1 PhD), and the remaining 4 comprised researchers, a software developer, and a translator. A convenience sampling was used to select participants, who were recruited from social contacts and posters on corridors at the University of Lisbon. No monetary reward was offered and the only information available was that the experiment would concern brainstorming.

## 4.2 Apparatus

The experiment was conducted in a laboratory room having five laptops with identical hardware (Intel Pentium M at 1.2 GHz, 1 GByte of RAM) and software specifications (Microsoft Windows XP SP2, .NET Framework 2.0), interconnected by a dedicated 100 Mbit/s Ethernet network. Keyboard sensitivity, desktop contents, display resolution, and brightness were controlled. Each computer had screen-recording software (ZD Soft Screen Recorder 1.4.3), and a web-camera (Creative WebCam Live!) affixed to the top of the screen. The client application of ABTool was installed on all five laptops.

## 4.3 Task

Participants completed practice and test tasks, both related to brainstorming. The practice task allowed participants to get familiar with ABTool. In the test task, participants were given a question and then asked to generate as many ideas as possible, by typing on the keyboard and by looking at the computer display. Speech and other forms of communication were disallowed to simulate a distributed work environment and to mitigate extraneous influences.

## 4.4 Design

A repeated measures design was chosen for the experiment. The independent variable was *device type* and every group of participants was under the influence of a control treatment, with immediate broadcast of ideas to the group, and an experimental treatment, with the opportunity seeker. The dependent variable, *group performance*, was calculated from the sum of the number of ideas produced by each user on the group per brainstorming session.

The order of exposure to the treatments and the brainstorming questions are depicted in Table 1. We note that, sometimes, session order is greater than two and that four questions were used, because we are reporting here a part of a larger experiment with two additional treatments, involving similar brainstorming tasks.

Table 1: Session order/brainstorming question per group and treatment. The questions were: A, how to preserve the environment; B, how to promote tourism; C, how to improve the university; and D, how to stimulate sports practice.

| | Groups | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Control | 1/C | 2/D | 4/C | 3/B | 1/B | 1/A | 2/C | 3/B | 2/B | 3/C | 1/A |
| Experimental | 3/B | 1/A | 2/B | 4/C | 3/C | 2/B | 3/A | 1/C | 1/C | 2/A | 3/B |

### 4.5 Procedure

A trial started when a group of participants arrived at the laboratory room. An introduction to this research was given and participants were informed on their privacy rights and asked to sign a consent form. Next, participants filled in an entrance questionnaire about gender, age, and occupation. Written instructions on the rules of brainstorming and on the ABTool application were then handed in to all participants and read out loud by the experimenter.

Participants were asked to carry out the practice task for 5 minutes, after which questions about ABTool were answered. The group then performed the test tasks in succession, each lasting for 15 minutes, with a brief rest period in between. At the end of the trial, answers were given to the questions participants had about this research, comments were annotated, and the experimenter gave thanks in acknowledgement of their participation in the experiment.

## 5 Results

Results are organised in three parts: firstly, an analysis of overall group performance, which is central to our research hypothesis; secondly, a decomposition of group performance into consecutive periods over a brainstorming session; finally, results from a post-hoc analysis based upon more fine-grained data, collected at the user level.

### 5.1 Group performance

Groups produced an average of 9.6% extra ideas per session when under the exposure of the opportunity seeker than under the control treatment, totalling 1251 vs. 1141 ideas for the 11 sessions (see Table 2).

Table 2: Number of ideas produced by groups under the two treatments.

| | Groups | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | Total | *M* | *SD* |
| Control | 152 | 83 | 133 | 91 | 264 | 77 | 48 | 53 | 66 | 104 | 70 | 1141 | 103.7 | 62.0 |
| Experimental | 192 | 108 | 113 | 117 | 258 | 77 | 68 | 61 | 76 | 116 | 65 | 1251 | 113.7 | 60.8 |
| Difference | 40 | 25 | −20 | 26 | −6 | 0 | 20 | 8 | 10 | 12 | −5 | 110 | 10.0 | 17.2 |

The Shapiro-Wilk normality test indicated that the normality assumption could not be accepted for both the control and experimental data distributions ($W = 0.795$, $p = 0.008$; and $W = 0.797$, $p = 0.009$, respectively). Therefore, we applied the non-parametric Wilcoxon signed-ranks test, which revealed a 3.7% probability of chance explaining the difference in group performance, $W_+ = 45.5$, $W_- = 9.5$.

We also analysed possible confounding influences from the questions or session order on group performance to see if there was a bias introduced by popular questions or a learning effect due to the nature of the repeated measures design. We applied the Wilcoxon signed-ranks test to both scenarios, which found no significant influences: $p > 0.205$ and $p > 0.343$, respectively.

Given this evidence, we can accept the hypothesis that group performance improved when groups were exposed to the opportunity seeker device in electronic brainstorming tasks with ABTool.

### 5.2 Group performance over time

Concerning the analysis of group performance through the duration of the brainstorming sessions, we broke down the 900 seconds that each session lasted into consecutive periods of 300, 150, and 30 seconds and counted the number of ideas put forward during each period. By following this approach we intend to highlight specific periods when one of the devices would enable better group performance. For example, a brainstorming session may be decomposed into at the beginning, when users usually have plenty of ideas, at the middle, and at the end, when users are typically more passive.

This decomposition is depicted in the top region in Fig. 7, which shows that in all three periods of 300 seconds groups produced more ideas with the opportunity seeker than with the control device. We obtained similar results at the 150 seconds level of aggregation (see middle region in Fig. 7). Finally, if we consider periods of 30 seconds (see bottom region in Fig. 7) then groups performed better with the opportunity seeker in 21 out of 30 cases. From the evidence collected, there seems to be no particular phase when group performance with the opportunity seeker could be considered worse than with the control device.

These results encouraged us to extend the analysis of group performance over time to other aggregation periods. We looked at the 26 divisors of session duration (in seconds), from counts of ideas generated in the two halves of a session (each lasting 450 seconds) down to the 900 aggregation periods of one

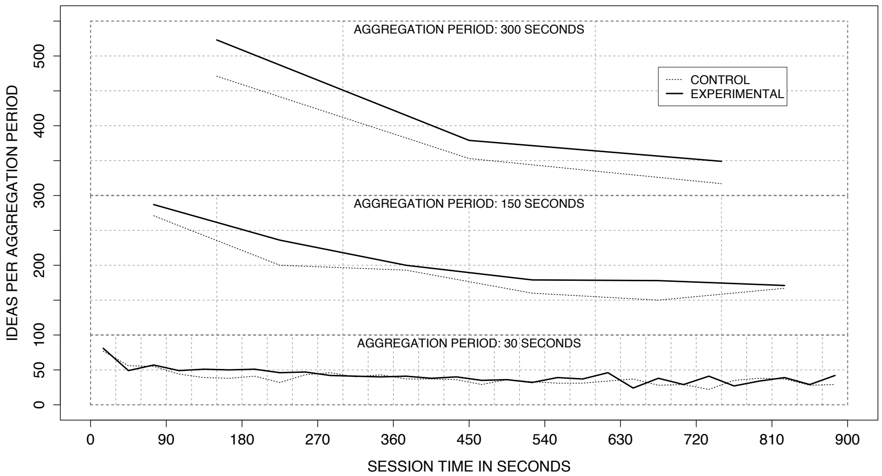


Figure 7: Group performance through the duration of the brainstorming sessions under the control and experimental treatments. *Top:* number of ideas per period of 300 seconds. *Middle and bottom:* same, considering periods of 150 and 30 seconds, respectively.

second each. Then, for all 26 aggregation periods we measured the percentage of cases over the duration of a session in which group performance was better, worse, and equal with the opportunity seeker compared to the control device.

In these circumstances, group performance with the opportunity seeker was better than with the control device in *at least* 40% of the cases, with an average of 68.3% ($SD = 18.9$), which contrasts with the percentage of cases in which it was worse: *at most* 40%, with a mean value of 24.4% ($SD = 14.0$). In other words, for all 26 aggregation periods considered, the opportunity seeker always had a higher proportion of cases over the session duration in which group performance was better than with the control device.

### 5.3 ***Post-hoc*** **analysis**

We also performed a post-hoc analysis based upon fine-grained data collected with ABTool to characterise the actual delivery of ideas and the performance of the users during the brainstorming sessions.

We considered the following variables: DLVR, deliveries of ideas per session; TBDL, seconds between consecutive deliveries; TIDEA, seconds to write an idea; PAUSE, seconds between a user submitting an idea to the group and restart typing; CIDEA, characters per idea; CHARS, total number of characters typed per user in a session; and DCHARS, total characters deleted per user per session.

Table 3 shows a summary of the results we obtained for all users that participated in the experiment, including descriptive statistics and the output of the Wilcoxon signed-ranks test, which we use here to prioritise the presentation of further details rather than to do null hypotheses significance testing. Thus, no family-wise corrections were made.

Table 3: Results of *post-hoc* analysis at the user level, ordered by $p$-value.

| | Control | | Experimental | | Difference | | Wilcoxon test | | |
|---|---|---|---|---|---|---|---|---|---|
| Variable | $M$ | $SD$ | $M$ | $SD$ | $M$ | $SD$ | $W_+$ | $W_-$ | $p$ |
| DLVR | 82.7 | 48.1 | 46.2 | 4.6 | −36.5 | 37.4 | 0.0 | 1540.0 | 0.000 |
| TBDL | 13.7 | 5.9 | 21.2 | 6.1 | 7.5 | 3.2 | 1540.0 | 0.0 | 0.000 |
| TIDEA | 25.7 | 17.3 | 21.5 | 11.8 | −4.2 | 12.9 | 422.0 | 1118.0 | 0.004 |
| PAUSE | 34.1 | 34.3 | 27.7 | 19.2 | −6.4 | 21.7 | 469.0 | 1071.0 | 0.012 |
| CHARS | 1044.8 | 511.2 | 1110.4 | 529.8 | 65.6 | 321.4 | 936.5 | 603.5 | 0.164 |
| CIDEA | 45.6 | 12.7 | 43.9 | 12.9 | −1.7 | 9.5 | 613.0 | 872.0 | 0.266 |
| DCHARS | 206.7 | 163.0 | 199.3 | 133.3 | −7.4 | 121.9 | 724.0 | 816.0 | 0.703 |

Starting with the DLVR variable, the experimental device reduced by an average of 44.1% the number of deliveries of group ideas that reached a user per session. This difference, from a mean value of 82.7 deliveries per session to 46.2, was due to each delivery having comprised a batch of 1.9 ideas on average ($SD = 1.2$), with up to 5 ideas in 99% of the cases and a maximum batch size of 9 ideas (happening only once), unlike when under the control treatment, in which new ideas were immediately broadcasted, one by one, to the group.

Another consequence of the opportunity seeker device, captured in variable TBDL, is that users had 54.7% more time, on average, to think about and type ideas without receiving new ideas from others, corresponding to uninterrupted periods with a mean duration of 21.2 seconds instead of 13.7 seconds with the control device.

The opportunity seeker trades up-to-date broadcasts of new ideas for less frequent deliveries of batches of ideas. This could have aggravated the alternation between doing individual work and attending to the group if, for instance, users had slowed down because of the apparent delays in group awareness updates or had become overloaded by the quantity of information in the batches.

In fact, variable TIDEA reveals that users needed a mean value of −16.3% of time to write an idea under the experimental treatment, corresponding to an average cut down of 4.2 seconds per idea when users typed their ideas without being interrupted with ideas from the other users. We also found, through variable PAUSE, that users switched 18.8% more rapidly, or 6.4 seconds faster, on average, from submitting an idea to the group to start typing the next idea, presumably reading ideas from others in between (see motivation near Fig. 2).

For the remaining variables in Table 3, results revealed small differences between the control and experimental treatments, thus likely explained by chance. The number of characters typed per user in a session, CHARS, was 6.3% higher, on average, with the opportunity seeker, influenced by the higher number of ideas produced (see Table 2), but balanced by slightly fewer characters per idea (CIDEA had a mean difference of −3.7%). Finally, the number of deleted characters, DCHARS, was 3.6% lower under the experimental treatment, on average.

## 6 Discussion

In this section, we elaborate on how users act when they receive new ideas from others and submit their ideas to the group, then we analyse the potential problem of some of the ideas not being delivered because of the buffering technique employed by the opportunity seeker, and, finally, we discuss the limitations of this study, in particular concerning the lack of a qualitative evaluation.

### 6.1 Validation of patterns of user activity

Earlier, we identified three patterns of user activity in brainstorming sessions with immediate broadcast of ideas, from the visual analysis of plots such as the one shown in Fig. 2. These patterns are important because they supply the basis for the model of user behaviour depicted in Fig. 3.

We now provide evidence for the first two patterns—that users typically do not stop typing when they receive new ideas from the other users and that they usually pause after putting forward an idea—based upon fine-grained data collected during the laboratory experiment.

On the one hand, in the first five seconds after the reception of new ideas from others, users continued typing their idea at a mean rate between 1.4 and 1.6 key presses per second ($SD$ between 0.7 and 0.8). On the other hand, after submitting an idea to the group, users almost stopped typing for at least five seconds, with a mean rate between 0.1 and 0.2 key presses per second ($SD$ between 0.2 and 0.3). This provides evidence to validate the two patterns mentioned above.

### 6.2 Undelivered ideas

One of the concerns of buffering ideas during brainstorming sessions, instead of immediately broadcasting them, is that the ideas submitted near the end of the session may not be delivered to some of the users. This can happen when a user is less productive than the others, either because s/he types very slowly or does not type at all due to lack of inspiration. As explained earlier, in these circumstances the opportunity seeker delays the delivery of new ideas from others, limited to a predefined quantity, until the user finally submits the idea to the group or until a timeout occurs, respectively.

Since it is undesirable to have undelivered ideas, we measured group production in each session with the opportunity seeker and subtracted from it the number of ideas from others actually received by each user. Table 4 shows that in 72.7% of the cases (or 40 sessions out of a total of 55) all ideas were delivered to the users and that in 20.0% of the times one or two ideas were not delivered; the remaining 7.3% were for cases with between three and seven undelivered ideas, each occurring only once.

Table 4: Sessions with undelivered ideas. Column 0 represents the special case in which all ideas were delivered to the users. No more than seven ideas remained to be delivered at the end of a session, and this happened only once.

| Undelivered ideas | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Number of sessions | 40 | 7 | 4 | 1 | 1 | 1 | 0 | 1 |

In other words, these data reveals that the users' natural work rhythm was rapid enough so that less than one idea ($M = 0.6$, $SD = 1.4$) was not delivered at the end of a session with the opportunity seeker, which seems reasonable.

### 6.3 Limitations

We had to accept several compromises for this study, most of them related to the absence of a qualitative analysis of both the users' ideas and the videos that were captured during the brainstorming sessions.

Firstly, we did not evaluate nor compare the quality of the ideas due to the subjective nature of this task and also because it would have required several evaluators, which have not been available so far. Then again, quantity is one the goals of brainstorming [11] and there is evidence that quality is positively linked to quantity [12].

Secondly, we did not investigate duplicate ideas, something that could be explicitly addressed in a qualitative analysis. The interest here would be to know if the buffering mechanism on the opportunity seeker artificially inflated the number of generated ideas by causing users to unknowingly submit ideas equivalent to those stored in the buffer but not yet displayed. However, with immediate broadcast of ideas users may not be able to keep up with the others, which might also lead to duplicate ideas. Thus, a comparison between the two conditions on this topic is appealing and its results could eventually help fine-tune the opportunity seeker.

Thirdly, we always used the same values for the parameters of the opportunity seeker: no more than ten ideas were presented at once and the inactivity period after which ideas would be delivered to the user was ten seconds. We could have experimented with other values (keeping in mind the objectives explained earlier, e.g., not filling up the computer screen with new ideas) but that would have increased the complexity of the experimental design beyond our current logistic capacity.

Fourthly, we faced many difficulties while examining the video feeds of the computer screen and the user's face. The purpose was to make observations related to the three patterns of user activity identified earlier: a) if users are able to attend to other users' ideas and write an idea simultaneously; b) if the pause in typing activity after the submission of an idea coincides with the user looking at others' ideas; and c) if periods of inactivity correspond to lack of imagination, distraction, or to engaged reading. However, the videos showed users who appear to be focused on the task and computer screen most of the time. Very occasionally, there was an outward reaction to reading an idea, e.g., a frown or smile. It was also infrequent to observe users acting distracted, for instance, staring somewhere else than the computer screen. Given this data, it was impossible to accurately distinguish when a user was reading ideas, pausing, or distracted, so we had to discard these data.

Finally, we did not assess the degree to which users experienced information overload, if any. There exist several techniques that could provide insight into this, such as physiological measures and self-assessments of mental workload [13], which could be applied in future experiments.

## 7 Conclusions and future work

We highlighted the need to apply Attentive User Interfaces to groupware systems and made contributions to address the group attention problem in synchronous electronic brainstorming settings.

Firstly, we presented an attentive device, the opportunity seeker, which applies buffering of group awareness information to mitigate information overload. The opportunity seeker considers the natural rhythms of group work to time the delivery of ideas with the situations in which users are most likely to benefit from them. Secondly, we showed how this device can be implemented on an electronic brainstorming tool and how task boundaries can be detected via keyboard activity. Thirdly, we provided evidence that the opportunity seeker can increase the work done by groups, and that the improvement amounts to 9.6% in the number of ideas produced in electronic brainstorming tasks.

In addition, results from a post-hoc analysis show that the opportunity seeker reduced the number of deliveries of ideas by 44.1% by combining ideas in small batches and that this translated into 54.7% more time for users to think about and type ideas without receiving new ideas from others. In these conditions, users were 18.8% faster in alternating between generating an idea, which they did in 16.3% less time, and reading other users' ideas.

We believe that the attentive device we propose in this paper provides benefits for today's and tomorrow's demands: on the one hand, even if the users in our experiment were not overloaded with information, the number of ideas produced was, nonetheless, higher; on the other hand, the opportunity seeker facilitates the creation of electronic brainstorming sessions with larger group sizes because it ensures that each user will be exposed to new ideas from others at his or hers own natural rhythm, thus automatically mitigating information overload.

As for future work, we are considering several research paths: one is to address the limitations presented earlier; another is to experiment with the opportunity seeker in other types of computer-mediated group tasks, especially in convergence tasks, such as negotiation; finally, we have plans to introduce an eye-tracker in future experiments.

## Acknowledgments

This work was supported by the Portuguese Foundation for Science and Technology, through project PTDC/EIA/67589/2006 and the Multiannual Funding Programme, and by the Departamento de Postgrado y Postítulo of the Vicerrectoría de Asuntos Académicos of the Universidad de Chile.

## References

1. Vertegaal, R.: Attentive user interfaces: Introduction. Communications of the ACM **46**(3) (March 2003) 30–33
2. Roda, C., Thomas, J.: Attention aware systems: Introduction to special issue. Computers in Human Behavior **22**(4) (July 2006) 555–556
3. Vertegaal, R., Shell, J.S., Chen, D., Mamuji, A.: Designing for augmented attention: Towards a framework for attentive user interfaces. Computers in Human Behavior **22**(4) (July 2006) 771–789
4. Fogarty, J., Ko, A.J., Aung, H.H., Golden, E., Tang, K.P., Hudson, S.E.: Examining task engagement in sensor-based statistical models of human interruptibility. In: CHI'05: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (April 2005) 331–340
5. Baudisch, P., DeCarlo, D., Duchowski, A.T., Geisler, W.S.: Focusing on the essential: Considering attention in display design. Communications of the ACM **46**(3) (March 2003) 60–66
6. Vertegaal, R., Weevers, I., Sohn, C., Cheung, C.: GAZE-2: Conveying eye contact in group video conferencing using eye-controlled camera direction. In: CHI'03: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (April 2003) 521–528
7. Jenkin, T., McGeachie, J., Fono, D., Vertegaal, R.: eyeView: Focus+context views for large group video conferences. In: CHI'05: Extended abstracts on Human factors in computing systems, New York, NY, USA, ACM Press (April 2005) 1497–1500
8. Pinelle, D., Gutwin, C.: Groupware walkthrough: Adding context to groupware usability evaluation. In: CHI'02: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (April 2002) 455–462
9. Bailey, B.P., Konstan, J.A.: On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. Computers in Human Behavior **22**(4) (July 2006) 685–708
10. McFarlane, D.C.: Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. Human-Computer Interaction **17**(1) (2002) 63–139
11. Osborn, A.F.: Applied imagination: Principles and procedures of creative problem-solving. Third edn. Scribner, New York, NY, USA (1963)
12. Briggs, R.O., Reinig, B.A., Shepherd, M.M., Yen, J., Nunamaker, J.F.: Quality as a function of quantity in electronic brainstorming. In: HICSS'97: Proceedings of the thirtieth Hawaii international conference on System sciences, Washington, DC, USA, IEEE Press (January 1997) 94–103
13. Wickens, C.D., McCarley, J.S.: Applied Attention Theory. CRC Press, Boca Raton, FL, USA (2008)