An Attentive Groupware Device to Mitigate Information Overload

Antonio Ferreira* and Pedro Antunes

Department of Informatics, University of Lisbon, Portugal
asfe@di.fc.ul.pt, paa@di.fc.ul.pt

Abstract We propose an attentive device for synchronous groupware systems to mitigate information overload. The opportunity seeker device leverages the users' natural alternation between doing individual work and attending to the group to dynamically manage the delivery timing and quantity of group awareness information that each user is exposed to. We describe how this device can be implemented in an electronic brainstorming tool and show its influence on the distribution of ideas to the users. Results from a laboratory experiment using this tool indicate that group performance increased 9.6% when compared to the immediate broadcast of ideas and a *post-hoc* analysis suggests that information overload was attenuated: users were subject to 44.1% less deliveries of ideas, which gave them 53.3% more uninterrupted time; users switched 12.2% faster from submitting an idea to start typing the next idea; and the time to write an idea was reduced by 14.8%.

1 Introduction

Attention management is increasingly important in our information-rich world as evidenced by the growing momentum of Attentive User Interfaces (AUI) in the field of Human-Computer Interaction (HCI) [1,2]. The prime motivation for AUI is the recognition that as the needs for information rise so do the costs of not paying attention to it. So, instead of assuming the user is always focused on the entire computer display, AUI negotiate the users' attention by establishing priorities for presenting information.

Most AUI research is grounded on single-user work and assumes user performance degrades with the number of simultaneous requests for attention. Therefore, researchers have enhanced 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 subject.

* Corresponding author.

Ferreira, A. and P. Antunes (2008) An Attentive Groupware Device to Mitigate Information Overload. Interactive Systems: Design, Specification, and Verification. Proceedings of the 2008 Conference on Design, Specification, and Verification of Interactive Systems, Kingston, Canada. T. Graham and P. Palanque. Lecture Notes in Computer Science, vol. 5136, pp. 29-42. Heidelberg, Springer-Verlag.

Firstly, the convergence of AUI and groupware systems poses new challenges to researchers due to differences in individual and group work:

- People working in a group are more occupied with requests for attention because they have to manage more information flows;
- Instead of doing a single extensive task, group members usually execute a series of intertwined tasks;
- Group members have to explicitly manage the trade-offs of attending to the group and doing individual work; and
- In group work the primary and secondary tasks are typically related and may both contribute to the shared goal.

Secondly, the current emphasis in AUI applied to groupware is still, to the best of our knowledge, on evaluating the enhanced input/output devices *per se*, e.g., the fluidity of movement or sudden brightness changes in videos [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 through devices such as radar views, multi-user scrollbars, and telepointers [8,9]. However, a problem with this trend is that it fails to recognise that sometimes more is less due to the limitations in 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 paying attention and becoming overloaded with information.

We argue that this problem is inadequately addressed by existing groupware awareness devices because they are designed having into consideration hardware limitations, e.g., decorators for telepointers to attenuate jitter effects due to network latency [10], but do not make any assumptions regarding the human attentive capacity. Furthermore, these devices require manual control of the type and quantity of group awareness information, e.g., via filters, thus penalising individual performance. On the other hand, the devices restrict the amount of information displayed to the user, which mitigates information overload.

This trade-off between the benefits of limiting group awareness information and manual intervention by the user sets the stage for introducing a conceptual attentive device for groupware systems to automatically adjust awareness information based upon each user's predicted state of attention, which we present in Sect. 3. In Sect. 4 we explain how this device can be implemented on an attentive brainstorming tool, and in Sect. 5 we describe a laboratory experiment to evaluate group performance with and without the attentive device, whose results are shown in Sect. 6. We conclude the paper in Sect. 7 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. The GAZE-2 system was developed to facilitate the detection of who is talking to whom in remote meetings [6]. It works by displaying video images of the users' faces on the computer display, which can be automatically rotated by intervention of eye-trackers placed in front of each user, e.g., 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 (but not eliminated). If the focus of interest suddenly changes, as sensed by the eye-tracker, the audio is again adjusted. Filters are also applied to the video images by manipulating their quality according to the angle of rotation (higher angles, lower quality).

eyeView explores the GAZE-2 ideas in the context of large meetings. It manipulates the size of video windows, arranged side-by-side, and the voice volumes of each user as a function of the current focus of attention [7].

These two groupware systems suggest that audio and video filters should be used 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 mentions technological evaluations through user questionnaires that measured the self-perception of eye-contact and distraction, as well as changes in colour and brightness during camera shifts [6]. A similar situation occurs with eyeView [7].

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 completion time, error rate, annoyance, and anxiety, and suggest that AUI should defer the presentation of peripheral information until task boundaries are reached [11]. 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 recommends that AUI should let users manually negotiate their own state of availability, except when response time for handling the interruptions is critical [12].

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 The opportunity seeker device

To address the group attention problem that we stated in the introduction—highlighting the need to mitigate information overload during computer-mediated group work—we devised an attentive groupware device, called the opportunity seeker, to dynamically manage the *delivery timing* and *quantity* of group awareness information based upon each user's state of attention.

There is a trade-off in managing the 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 for interrupting the user. Following Bailey and Konstan [11], 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).

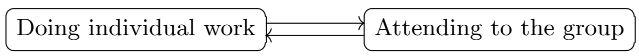


Figure 1: Natural task switching during group work.

Conceptually, the opportunity seeker has a queue for storing group awareness over time and this information should only displayed to the user when s/he is likely *not* doing individual work. Furthermore, a limit may be enforced to the quantity of information delivered at each opportunity if the rhythms of the user and the group differ too greatly, to avoid overloading the user.

4 An attentive brainstorming tool

We implemented the opportunity seeker device on ABTool, a custom-made electronic brainstorming tool with built-in sensors of user performance, to dynamically manage the delivery timing and quantity of ideas displayed to each user over brainstorming sessions. In electronic brainstorming users can submit ideas in parallel and 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 flow of ideas, and may even become distracted by it, thus causing information overload.

A major challenge in applying the opportunity seeker to ABTool was to detect task switching during electronic brainstorming activity. Theoretically, the rules of brainstorming [13] 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. From a practical viewpoint, we analysed data from ABTool's logs of activity running with immediate broadcast of ideas (see sample and comments in Fig. 2), from which three patterns of user activity emerged: firstly, 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; secondly, users typically paused after putting forward an idea, presumably to keep up with the group; and thirdly, we found numerous periods of time with no typing activity (not shown in Fig. 2).

Based upon this evidence, 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 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).

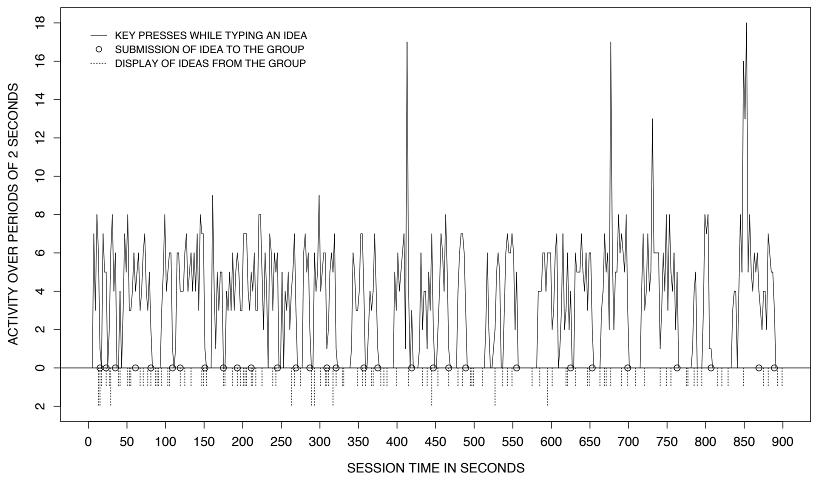


Figure 2: User and group activity during a brainstorming session with ABTool, with instant broadcast of ideas to everyone on the group. Above the X-axis are aggregated counts of user key presses. The spikes occurred when the user pressed the delete or cursor keys. The circles on the X-axis show when 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.

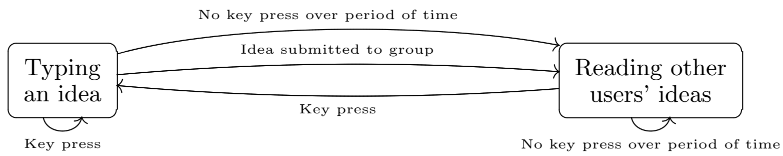


Figure 3: Model of user behaviour assumed by the opportunity seeker on ABTool.

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.

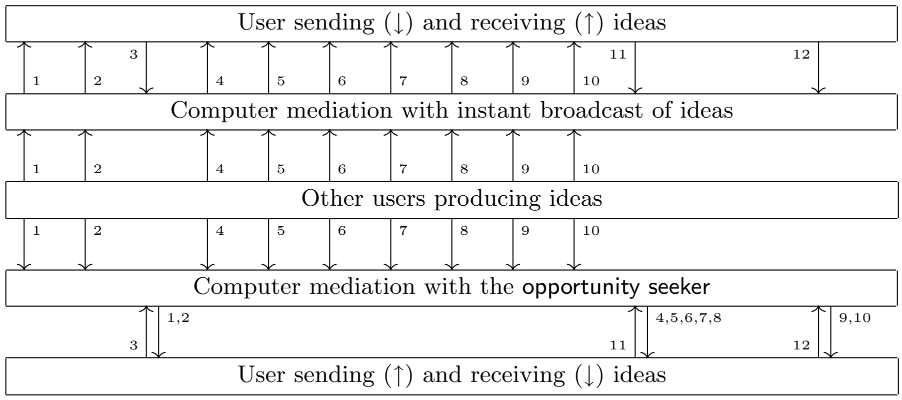


Figure 4: Simulation of group and user activity during a brainstorming session with immediate broadcast of ideas (*upper region*) and with the opportunity seeker (*lower region*). In both cases the user produces three ideas (numbered 3, 11, and 12) but the exposure to the nine ideas s/he received from 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.

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 built on top of 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 de-serialised 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 in 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 together with 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 on ABTool 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 queues, one per user, containing ideas that have been put forward by the other users on the group. The queue 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.

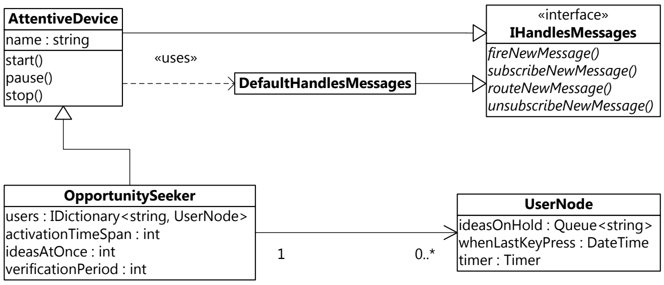


Figure 5: Class diagram showing details of the opportunity seeker on ABTool.

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.

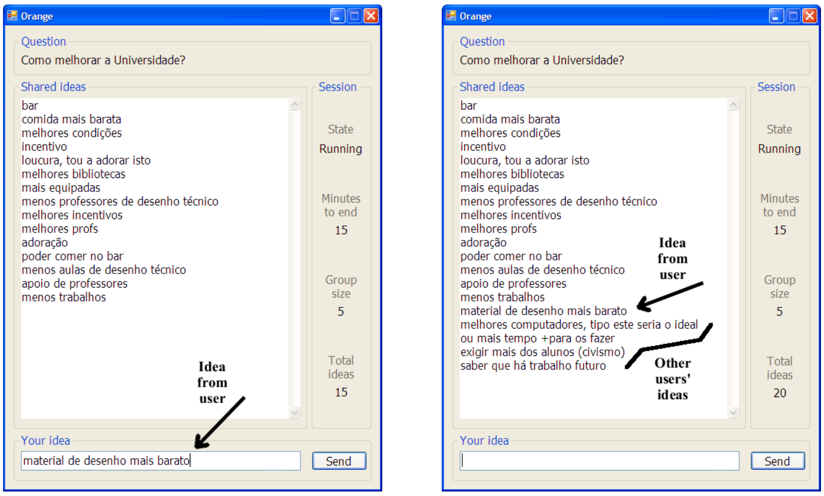


Figure 6: Opportunity seeker managing the delivery of ideas on ABTool. *Left:* While typing an idea, the user receives no new ideas from the group. *Right:* When the user submits an idea to the group, new ideas from others are displayed.

5 Laboratory experiment

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

5.1 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.

5.2 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 webcamera (Creative WebCam Live!) affixed to the top of the display. The client application of ABTool was installed on the five laptops and the server was installed on an additional laptop used by the experimenter.

5.3 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.

5.4 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 (CT)—with immediate broadcast of ideas to the group—and an experimental treatment, with the opportunity seeker (OS). 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.

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

	Group										
	1	2	3	4	5	6	7	8	9	10	11
CT	1/C	2/D	4/C	3/B	1/B	1/A	2/C	3/B	2/B	3/C	1/A
OS	3/B	1/A	2/B	4/C	3/C	2/B	3/A	1/C	1/C	2/A	3/B

The order of exposure to the treatments and the brainstorming questions used with the 11 groups 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.

5.5 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.

6 Results

Results are organised in three parts: we begin with an analysis of overall group performance, which is central to our research hypothesis; we then decompose group performance in consecutive periods over a brainstorming session; finally, we show results from a *post-hoc* analysis based upon more fine-grained data.

6.1 Group performance

Groups produced an average of 10.0 extra ideas per session ($SD = 17.2$), +9.6%, when under the exposure of the opportunity seeker (OS, $M = 113.7$, $SD = 60.8$) than under the control treatment (CT, $M = 103.7$, $SD = 62.0$). A total of 1251 ideas were put forward with the OS versus 1141 with the control device (see

Table 2: Number of ideas per group and treatment.

	Group											
	1	2	3	4	5	6	7	8	9	10	11	Total
CT	152	83	133	91	264	77	48	53	66	104	70	1141
OS	192	108	113	117	258	77	68	61	76	116	65	1251

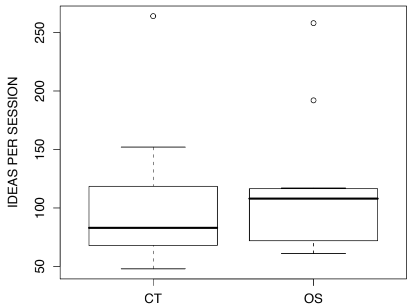


Figure 7: Group performance under the influence of the control (CT) and experimental (OS) treatments.

Table 2). Figure 7 further shows that the difference between treatment medians was 25 ideas per session (108 vs. 83).

The Shapiro-Wilk normality test indicated that both data distributions differed significantly from a normal distribution; therefore we applied the non-parametric Wilcoxon signed-ranks test, which revealed a significant 3.7% probability of chance explaining the difference in group performance.

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. In other words, group performance can increase by managing the delivery timing and quantity of group awareness information displayed to the users.

6.2 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 using this approach we intended to highlight specific periods when one of the devices would enable better group performance. For example, a brainstorming session may be divided 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 division is actually depicted in the top region in Fig. 8, which shows that in all three periods of 300 seconds groups produced more ideas with the opportunity seeker than with the control device. This outcome is reinforced by similar results at the 150 seconds level of aggregation (see middle region in Fig. 8).

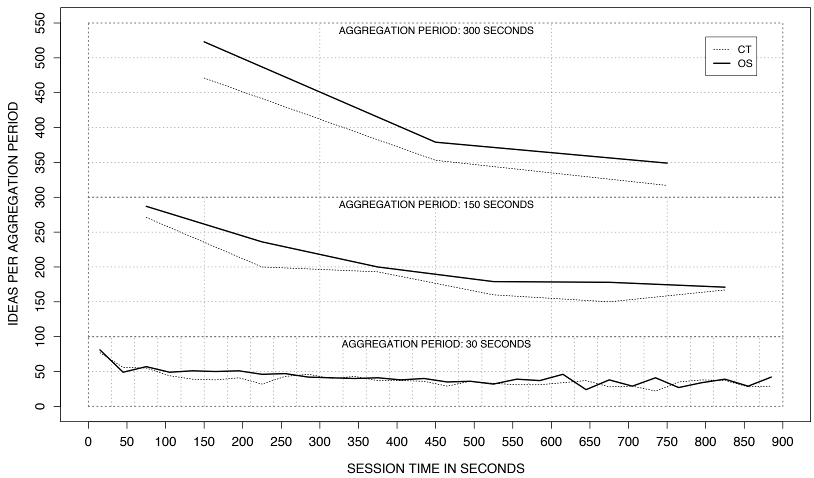


Figure 8: Group performance through the duration of the brainstorming sessions under the control (CT) and experimental (OS) treatments. *Top:* number of ideas per period of 300 seconds. *Middle and bottom:* same, considering periods of 150 and 30 seconds, respectively.

Finally, if we consider the count of ideas collected over consecutive periods of 30 seconds (see bottom region in Fig. 8) then group performance with the opportunity seeker is better in 21 out of 30 cases than with the control device.

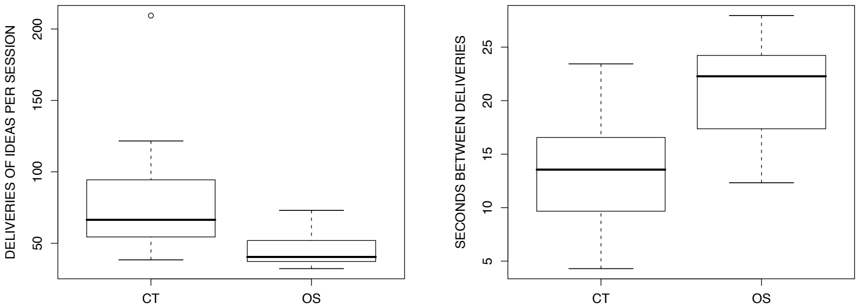
We do not provide more statistics for this type of analysis because its meaning would be attached to the choice of periods, which depends on the context. Instead, we note that there seems to be no particular phase when results with the opportunity seeker could be considered worse than with the control device.

6.3 *Post-hoc* analysis

We also performed a *post-hoc* analysis comprising the influence of the opportunity seeker in the delivery of ideas and a fine-grained study of user performance in terms of task switching time and individual work.

The opportunity seeker device reduced the number of deliveries of group ideas that reached a user in each session by 44.1%, from an average of 82.7 ($SD = 49.1$)

to 46.2 ($SD = 13.6$). Figure 9a shows more details. This was possible because each delivery comprised a batch 1.9 ideas on average ($SD = 1.2$), with up to 5 ideas per batch in 99% of the cases, unlike with the control device, in which new ideas were immediately broadcasted, one by one, to the group.



(a) (b)

Figure 9: Characterisation of idea deliveries under the control (CT) and experimental (OS) treatments.

Users also had 53.3% more time to think about and type ideas without receiving new ideas from others: an average of 20.8 seconds with the opportunity seeker ($SD = 5.2$) vs. 13.6 ($SD = 5.8$) with the control device (see Fig. 9b).

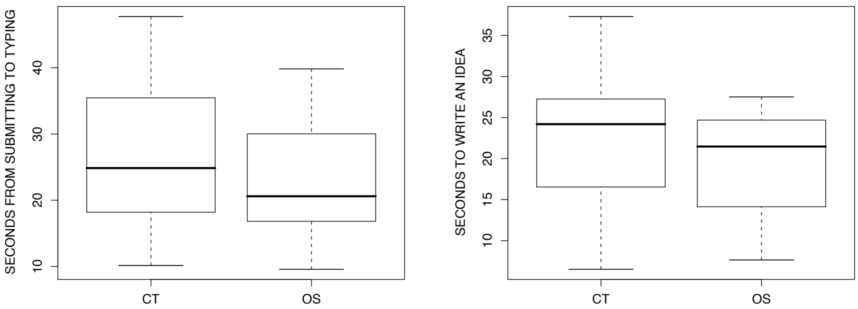
The opportunity seeker trades up-to-date group awareness for less frequent deliveries of batches of information. This could have aggravated the alternation between doing individual work and attending to the group if, for instance, users had slowed down because of the apparent delays in group awareness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

In fact, users switched 12.2% more rapidly from submitting an idea to the group to start typing the next idea, presumably reading ideas from others in between: 23.6 seconds per idea ($SD = 10.3$) vs. 26.8 ($SD = 11.9$), on average (see Fig. 10a). We also found that, with the OS device, users needed an average of 19.4 seconds ($SD = 6.4$) versus 22.7 ($SD = 8.6$), −14.8% of time, to type an idea (see Fig. 10b).

This evidence suggests that the opportunity seeker on ABTool mitigated information overload by leveraging the users' natural rhythms for doing individual work and attending to the group to manage the delivery of ideas.

7 Conclusions and future work

We highlighted the need to apply Attentive User Interfaces beyond single-user systems and to multi-user systems, e.g., due to the differences in individual and group work, and made contributions to address the group attention problem.



(a) (b)

Figure 10: Aspects of user performance under the control (CT) and experimental (OS) treatments.

Firstly, we devised an attentive groupware device, the opportunity seeker, that acknowledges the users' natural alternation between doing individual work and attending to the group, and manipulates the delivery timing and quantity of group awareness based upon the user's predicted state of attention. 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 device 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 53.3% more time to think about and type ideas without receiving new ideas from others. In these conditions, users were 12.2% faster in alternating between generating an idea, which they did in 14.8% 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 implement the opportunity seeker on other types of computer-mediated group tasks, such as instant messaging; another path is to analyse the videos we have captured with the screen recorder and the web-camera during the brainstorming sessions, to assess our assumptions about the users' focus of attention in this context, so far based solely upon activity logs; finally, we have plans to gather more fine-grained data (compared to video analysis) by introducing 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.

References

1. Vertegaal, R.: Attentive user interfaces: Introduction. Communications of the ACM **46**(3) (2003) 30–33
2. Roda, C., Thomas, J.: Attention aware systems: Introduction to special issue. Computers in Human Behavior **22**(4) (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) (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 (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) (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 (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 (2005) 1497–1500
8. Raikundalia, G.K., Zhang, H.L.: Newly-discovered group awareness mechanisms for supporting real-time collaborative authoring. In: AUIC'05: Proceedings of the sixth Australasian conference on User interface, Sydney, Australia, Australian Computer Society (2005) 127–136
9. Gutwin, C., Greenberg, S.: A descriptive framework of workspace awareness for real-time groupware. Computer Supported Cooperative Work **11**(3) (2002) 411–446
10. Gutwin, C., Benford, S., Dyck, J., Fraser, M., Vaghi, I., Greenhalgh, C.: Revealing delay in collaborative environments. In: CHI'04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (2004) 503–510
11. 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) (2006) 685–708
12. 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
13. Osborn, A.F.: Applied imagination: Principles and procedures of creative problem-solving. Third edn. Scribner, New York, NY, USA (1963)