TeamMeetingArranger: A Less Disruptive Way of "Do you have a minute?"

Moritz Maleck[0009-0006-5551-8386], Tom Gross[0000-0001-8353-7388]

Human-Computer Interaction Group, University of Bamberg, 96045 Bamberg, Germany
hci@uni-bamberg.de

Abstract. Interruptions in the workplace can disrupt productivity, especially in team environments where collaboration is essential. Existing Interruption Management Systems often focus on individual interruption management and not on team dynamics, leading to inefficient workflows. We introduce TeamMeetingArranger, a novel system that dynamically schedules meetings based on real-time cognitive load assessments using eye tracking. The system identifies optimal times for meetings when the cognitive load of users is low. This reduces the disruptive impact of interruptions and improves individual and team productivity. TeamMeetingArranger provides a solution that can be seamlessly integrated into work environments to enhance productivity and workplace satisfaction.

Keywords: Effortless Coordination; Team Meetings; Arrangements; Interruptions; Interruption Management Systems; Cognitive Load; Eye Tracking.

1 Introduction

Interruptions are a common challenge in both individual and team work environments, potentially negatively impacting productivity [1, 2]. The nature of interruptions and the cognitive load experienced at the time of disruption play a critical role in determining their effect on performance [2-4]. To address this, we present the TeamMeetingArranger, a concept and system designed to reduce the disruptive nature of meetings by aligning meeting times with periods of low cognitive load among team members.

Our approach leverages real-time cognitive load assessment using eye-tracking technology, specifically measuring pupil diameter [5-7], to monitor and determine optimal meeting times. This allows the system to adjust meeting schedules dynamically based on each participant's current mental workload. By reducing interruptions during periods of high cognitive load, we aim to improve both individual focus and team collaboration, particularly in open-plan offices and hybrid work environments.

In this paper, we present TeamMeetingArranger. The key contributions of this work include: A novel approach for scheduling meetings based on real-time cognitive load data using eye tracking to minimise the disruptive nature of interruptions at work; and a flexible and scalable architecture that can be easily integrated with existing workplace tools and adapted for various team settings.

2 Related Work

The background and related work includes interruptions in general and in the workplace, social behaviour for the coordination in teams, cognitive load, and eye tracking.

Interruptions disrupt primary tasks and can result in time-consuming resumption efforts [1]. These can be split into few key stages—firstly, the interruption lag when a user is alerted to the secondary task, then engaging in the secondary task, and finally resuming the primary task after the interruption [8]. Even though existing research has predominantly focused on the negative aspects, interruptions can also have positive effects like faster completion of simple tasks or moments of incubation to foster creativity [1]. Whether an interruption is disruptive on the performance on return to the main task depends on the nature (i.e., the similarity) and complexity of an interruption [3].

Yet, in work settings interruptions often increase task completion time and can affect task accuracy and efficiency [1]. Especially in team settings collaborative processes can get delayed and the performance of not only the individual but the entire team may be affected [1, 9]. Interruptions can reduce productivity and well-being for office workers [2]. Tools for notification management help to reduce the interruptions by providing notifications in the right moment, but require considerable technical effort [10, 11].

Social cues are used to communicate openness for interruptions, such as opening or closing the door [2]. The applicability of such social cues is limited when it comes to the trending concept of open-plan offices. In such, employees report lower perceptions of workspace effectiveness, attractiveness, and satisfaction [12]. Therefore, a mix of open and private workspaces is proposed to meet the needs for both collaboration and individual, focused work [12]. From a positive side, interruptions due to open-plan offices can enhance social connections between individuals [1, 12].

Since COVID-19, working from home is an emerging trend [13, 14]. Studies mostly report a positive impact on productivity and performance, mostly in non-collaborative work settings [14]. But also the nature of meetings differs—more larger group meetings than one-to-one meetings [15]. Especially team performance declines for teams working mainly from home [16]. Reasons for home office, or hybrid solutions, might be found in the self-influence on the number and type of interruptions—e.g., by disabling notifications while focus-work—since repeated switching of attention required during work intrusions depletes employees' self-regulatory resources which has a negative impact on job satisfaction [17]. Moreover, employees have the possibility to deliberately shape their work environment in order to enhance concentration for focus work [18]. On the other hand, non-work interruptions are increasing [19]. Home office also brings new challenges in collaboration and innovation [13]. Thus, instead of leveraging on home office, we propose to improve workplace conditions.

Interruptions have an impact on cognitive load and vice versa. For example, interruptions increase cognitive load, as through the resumption lag where after an interruption one has to determine where the previous task had been interrupted to figure out the next steps [1]. Moreover, interruptions have been found to be more harmful during phases of high cognitive load [2, 4].

Eye tracking, particularly the measurement of pupil diameter, has proven effective for assessing cognitive load. Unlike other eye-tracking features like saccades and fixations [20, 21], pupil diameter is a reliable indicator of cognitive load since it cannot be consciously controlled [5]. Studies show that pupil dilation reflects not only changes in light but also cognitive processes [5, 21-23]. Yet, the pupil diameter can also be influenced by external factors like emotions, making controlled environments essential for accurate measurement of cognitive load [5]. Various systems already use the pupil diameter for real-time cognitive-load scenarios [6, 7, 24].

Cognitive load has already been used in systems to manage interruptions, optimising the timing of interruptions to minimise disruption. By the usage of workload-aligned task models, right moments for interruptions can be determined [4]. [2] used smartphones mounted at each office door to display a red or green background, signalling whether the office worker is interruptible or not, by inferring cognitive load using consumer wearables. [7] identify low-workload moments by analysing the pupil diameter of users and tested their method in a single user scenario with an email-answering task with interruptions during working. Their method showed to improve performance compared to interrupting at random times.

Intelligent Interruption Management Systems in team-based work, especially in but not limited to open-plan offices, can help to foster productivity. The measurement of pupil diameters is a non-invasive method to determine the cognitive load of users, that can be easily integrated in existing work settings. To the best of our knowledge, we are the first to use cognitive-load based eye tracking for managing interruptions due to meeting requests in team-based work.

3 TeamMeetingArranger Approach

Interruptions—depending on their nature and complexity—during work can negatively affect the productivity; and interruptions during high cognitive load are more harmful than during low cognitive load. Technical solutions can help here: We introduce *TeamMeetingArranger*, an approach for supporting employees within a team for determining optimal time frames for meetings with other employees.

TeamMeetingArranger suggests the optimal time for requested meetings based on the cognitive load of the meeting participants; once all meeting participants are below a certain cognitive-load threshold, the system notifies all participants and asks them to meet other team members. The cognitive load of the users is determined by the continuous assessment of the pupil diameter, a real-time indicator for cognitive load. For this, all users are equipped with an eye tracking device. The system consists of a centralised logic building upon global knowledge, and clients for each member within the team.

3.1 Determining Cognitive-load Level for Individual Team Members

The current cognitive-load level of users is determined based on the pupil diameter measured by a connected eye tracker. Each team member can request meetings with other team members or start and end meetings.

Cognitive-load Level Determination and Dynamic Threshold. Eye tracking reveals real-time mental processes, and especially the pupil diameter has been found to be a reliable indicator for the cognitive load. With high cognitive load, the pupil diameters increases. For the determination of the cognitive load level of a user, the recorded pupil graph of the last *n*=60 minutes is used as reference (where *n* shall be configurable). The cognitive-load level is then calculated as follows, where *p* is the current pupil diameter, min the minimum pupil diameter during the last n minutes and max the maximum:

$$CLLevel = \frac{p - min}{max - min}$$

This way, the cognitive-load level is a value between zero and one; the greater the value is, the higher the cognitive-load is.

Dynamic Threshold for Meeting Availability. To initiate meetings, a threshold for the maximum allowed cognitive-load level is defined. Instead of using a fixed threshold, we introduce a dynamic threshold with respect to the remaining working time of a user. That means, with longer time towards the planned quitting time, the threshold is lower; linearly increasing towards the quitting time. Therefore, users are required to provide their planned quitting time. This way dwindling remaining working times are taken into account and thus the probability is increased that a requested meeting can take place, even if the fixed threshold would not actually allow it.

3.2 Phases and States While Using TeamMeetingArranger

We defined four phases for the users of the TeamMeetingArranger in a defined order (cf. Figure 1). These phases are used by the central logic for arranging meetings, but are also available to the team members to provide awareness information of the availability or non-availability of the other members.

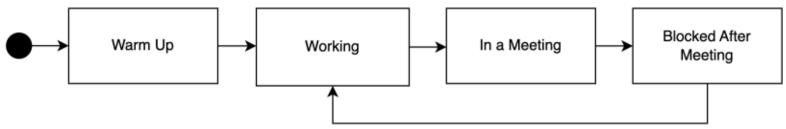

Fig. 1. The four phases for each user of the TeamMeetingArranger as program lifecycle during runtime, i.e., the working day.

Warm Up. After starting the TeamMeetingArranger and having connected to a team users start with the *warm-up* phase. The length of this phase is a fixed and configured value. During it, no meetings will be initiated. This allows the continuous measurement of the pupil diameter as indicator for the cognitive load. The stored pupil data are then used to make informed decisions after the end of the warm-up phase and thus with the beginning of the regular *working* phase. But also apart from data gathering, users benefit from the ability to start their working day without interruptions too soon.

Working. The *working* phase denotes the phase when users are doing regular screen-based work and are potentially open for interruptions in terms of meeting requests. Being in this phase with a high cognitive-load level, no meetings with other users are initiated. On the other hand, having a low level, requested meetings are only triggered if the condition is true for all of the meeting members. During this phase, a user can request meetings with other team members.

In a Meeting. If a user is a potential member of a requested meeting (being irrelevant who requested it) and all team members are below their individual dynamic cognitive-load level threshold, a meeting is initiated by informing all clients using a push notification shown at the top right of the screen. The push notification asks users to meet the other members of the meeting. During this phase, all meeting members are unavailable for further active meetings. For ending a meeting it is sufficient if one of the members confirms the end of the meeting.

Blocked After Meeting. After a meeting, users are temporarily blocked for the initiation of other requested meetings. The length of this phase is configurable.

3.3 Centralised Logic Building Upon Global Knowledge for Arranging Meetings

TeamMeetingArranger uses a centralised logic with global knowledge. All users are known within this construct. A list is maintained with all online and active users, all requested and active meetings. Based on a configured interval, all requested meetings are revised repetitively by checking each meeting members' transmitted cognitive-load level against the dynamic threshold. If for a certain user multiple meetings would be possible, the meeting with the oldest request timestamp is decided for. This way, fairness is ensured and by using a FIFO (first in first out) queue.

In the next section we provide the user interaction concept of the TeamMeetingArranger.

4 TeamMeetingArranger User Interaction Concept

The TeamMeetingArranger provides a web application for the clients. When opening the application in the browser, the user enters the address of the team server—that is, the address of the coordinator (cf. Figure 2 A). The user also provides a unique name (that is further used as identifier by the coordinator) and their planned quitting time, that is used for the dynamic cognitive-load threshold for the interruptibility for meetings during the working status. Before connecting, push notifications have to be allowed so that the user can be notified by the coordinator when a meeting is up.

Once connected, the application informs about the *warm-up* phase and the remaining time until meetings can be scheduled (cf. Figure 2 B). During the warm-up phase, the client already continuously measures the pupil diameter for later informed decisions on the initiation of meetings by the coordinator.

After the warm-up phase, the user interface is automatically replaced by a screen informing about the start of the *working* phase, and further showing other team members and their states as well as allowing arranging meetings and managing already pre-arranged meetings (cf. Figure 2 C). Meetings can be requested with at least one other team member (cf. Figure 2 D).

Once the coordinator initiates a meeting, meeting members are informed by a push notification, asking to meet the other team members. When clicking on the notification, the web application then provides details on the arranged meeting such as the members to meet and the organiser (cf. Figure 2 E). At the end of the meeting, one of the members should confirm the end by clicking the dedicated button. All members are then temporarily blocked for further meetings with the state *blocked after meeting*.

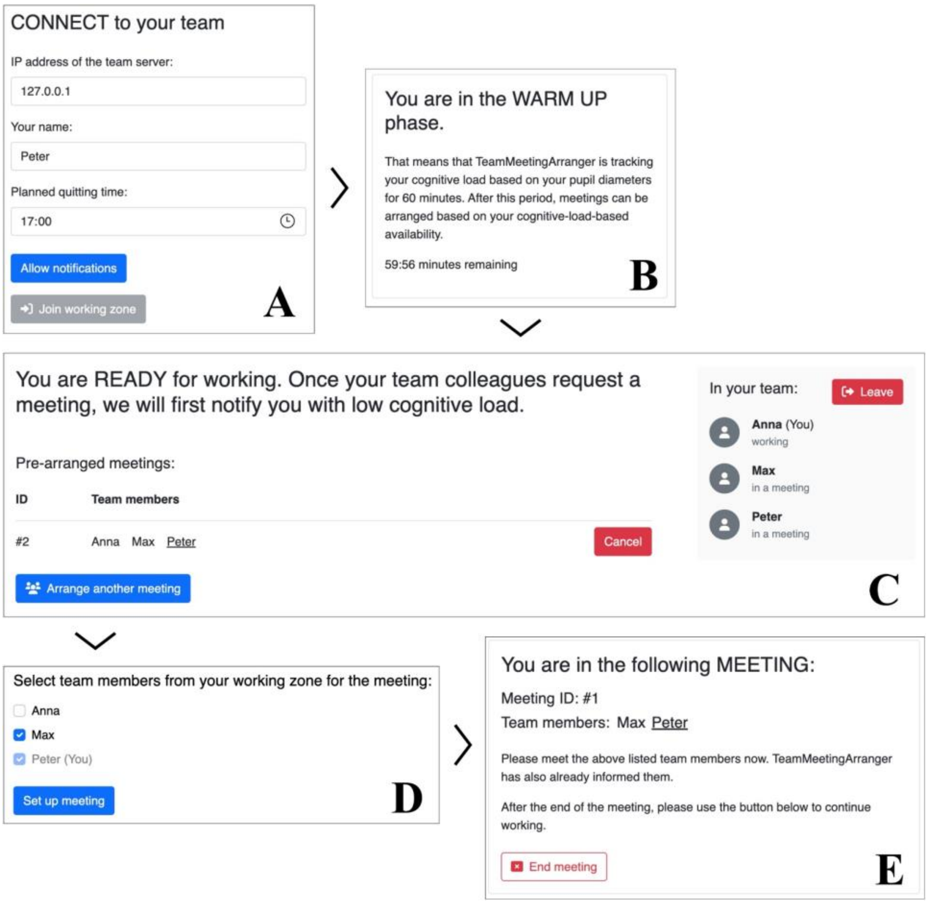

Fig. 2. Screens of the TeamMeetingArranger, implementing the four phases *Warm Up*, *Working*, *In A Meeting* and *Blocked After Meeting*.

In the next section we introduce the implementation of the TeamMeetingArranger system, containing the client and the coordinator server.

5 TeamMeetingArranger Implementation

In this section, we describe the implementation and functionality of the TeamMeetingArranger system. The system is comprised of a centralised coordinator with global knowledge and a client-side web application (cf. Figure 3). The client also integrates with an eye-tracker connector service, which links to an eye-tracking device and supplies real-time pupil diameter data.

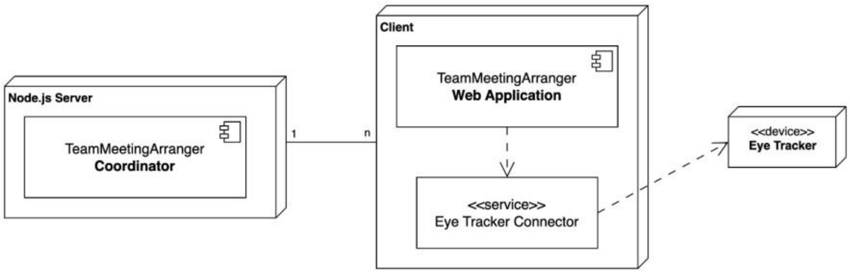

Fig. 3. System overview of the TeamMeetingArranger.

5.1 Web Application and Eye Tracker Connector Service for Clients

The client implementation includes a web application and is central to the users. It connects to the eye tracker for the measurement of the pupil diameters and to calculate cognitive-load levels, transferring these to the coordinator.

Web Application. We implemented the user interaction concept as described in the previous section. We developed it with JavaScript using the Svelte framework (version 3.48.0).

Eye Tracker Connector Service and Cognitive-Load Level Calculation. The web application uses a dedicated eye-tracker connector service to retrieve pupil diameter measurements. We implemented the service in Python 3.8.10, running on Windows 10, and interfacing with a Tobii Pro Spectrum eye tracker. Although the current prototype is designed for use with the Tobii eye tracker, the architecture is easily adaptable to accommodate other eye-tracking devices.

Given that TeamMeetingArranger operates as a real-time application with stringent requirements for low processing times, particularly for the coordinator with multiple clients connected, the entire pupil size graph is not utilised for computational tasks. Instead—to optimise performance—the service streams pre-processed data chunks to the web application, each containing the minimum, maximum, and mean pupil diameter values for the current chunk (cf. Figure 4). This segmentation of the real-time pupil data is necessary to accommodate resource-intensive processing steps, including filtering erroneous measurements, imputing missing values, and performing interpolation.

Once a new chuck is retrieved from the eye tracker connector, the new cognitive-load level is calculated. For this, a rolling window as a set of m=60 retrieved chunks (with one minute of eye tracking data per chunk) is used. Each new chunk of data is appended to the list *chunks[]*. If the list exceeds m items, the oldest entry is removed.

The maximum and minimum pupil diameters are determined from the stored chunks. The cognitive-load level is then the mean of the current chunk minus the minimum value divided through the range (i.e., the maximum pupil diameter minus the minimum). This results in a value between 0 and 1, where higher values indicate greater cognitive load. The window size can be adjusted by modifying *m*. The concrete algorithm is listed below.

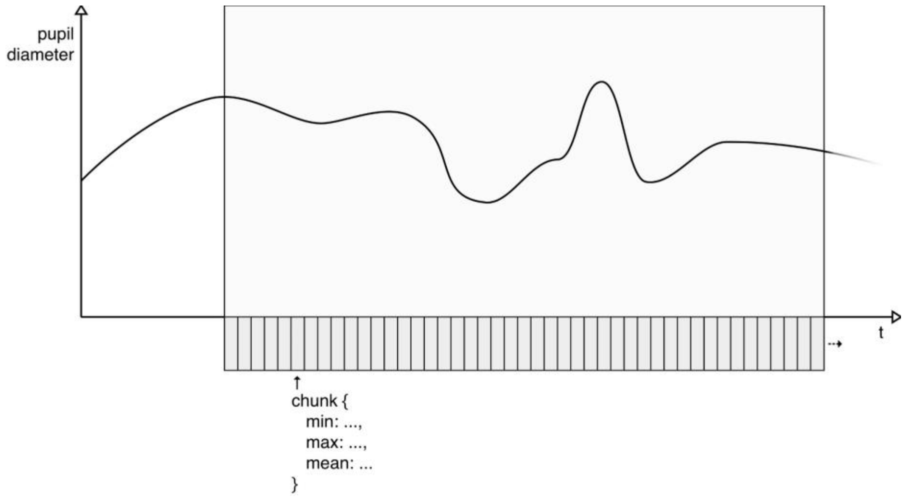

Fig. 4. Schematic visualisation of the eye tracking data provided by the eye tracker service to the client. The service streams chunks with relevant information to the client.

```
m ← 60
CLLevel ← undefined
chunks[] ← Empty list for pupil diameter chunks
chunk_retrieved(c):
  chunks[] ← Push c
  If |chunks| > m:
    chunks[] ← Remove first item
    maxPD ← max(max values of each item of chunks[]
    minPD ← min(min values of each item of chunks[])
    CLLevel ← (c.mean - minPD) / (maxPD - minPD)
```

5.2 Coordinator for Determining Possible Meetings

The coordinator is the meeting arrangement mechanism behind the TeamMeetingArranger. It receives cognitive-load levels from the clients. Based on these data, the coordinator makes informed decisions about the fulfilment of meeting requests by the clients. The coordinator is deployed on a server using Node.js (version 21.4.0).

Determining Possible Meetings. The coordinator repetitively identifies potential meetings based on user availability and cognitive load constraints. For this, a list of all requested meetings without duplicates is initiated. Only users who are currently in the *working* state are considered for new meetings.

Then, for each meeting of requested meetings, the logic iterates over its members. If (i) all members are in working state, (ii) their cognitive load level is below a dynamically calculated threshold, and (iii) none of the users is already assigned to one of previously iterated meetings, the meeting is added to the list of possible meetings.

This process ensures that only users in a working state with an acceptable cognitive load level are considered for possible meetings. The coordinator will inform the members of determined possible meetings to start the meeting. The algorithm of the above described logic is provided below.

```
determine_possible_meetings:
  Requested_meetings[] ← List with all requested meetings
                         without duplicates
  Working_users[] ← List with users with 'working' state
  Possible_meetings[] ← Empty list for possible meetings
  For each meeting M of Requested_meetings[]
    For each member U of M.members[]
      If Working_users[] contains U
      and U.CLLevel <= dynamicCLThreshold(U)
      and U is not member in any of Possible_meetings[]:
        Possible_meetings[] ← Push M
  Return Possible_meetings[]
```

Dynamic cognitive-load threshold. Instead of using a fixed cognitive-load threshold the coordinator depicts on a dynamic threshold based on the remaining working time of individual users. For this, a base working period is defined with *w*=480 minutes (i.e., eight hours). Based on a base threshold *CLThreshold* and the remaining working time with respect to *w*, a dynamic threshold is calculated, so that the threshold linearly increases towards the planned quitting time. This way, a higher tolerance for interruptions is applied if the remaining time for meeting requests would not allow initiation anymore. The algorithm is provided below.

```
dynamicCLThreshold(U):
  w = 480
  d ← min(remainingWorkingTimeInMinutes, w)
  Return CLThreshold + ((100 – CLThreshold) * (d / w))
```

5.3 Communication Between Coordinator and Clients

TeamMeetingArranger contains multiple clients and a centralised coordinator with global knowledge. Once clients register at the coordinator, bi-directional data channels are used for communication.

Client-Coordinator Architecture for Managing Meeting Requests. For managing meeting requests and tracking cognitive load, we implement a client-coordinator architecture (cf. Figure 5). Clients communicate with a central coordinator via dedicated WebSocket data channels in both directions: client-to-coordinator and coordinator-to-client. This architecture supports real-time meeting management, where clients can dynamically interact with the coordinator to set up or modify meetings.

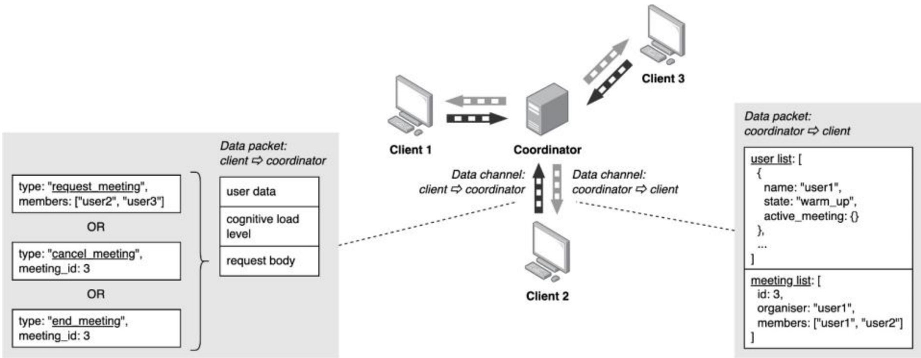

Fig. 5. TeamMeetingArranger consists of one coordinator and at least one client. The coordinator and the clients communicate via bi-directional data channels.

Client-to-Coordinator Communication. Clients send different types of data packets to the coordinator, each containing user-specific information, cognitive load level, and a request body. Three main types of requests are possible: *Request Meeting*: Clients initiate meetings by specifying the members. *Cancel Meeting*: Clients can request to cancel requested meetings. *End Meeting*: At the end of an initiated meeting, meeting members end the meeting using the dedicated request type.

Coordinator-to-Client Communication. The coordinator manages meeting information and client states, which it communicates to clients through the coordinator-to-client channel. The response includes: *User List*: Containing all users including their names, states, and—if applicable—active meetings. *Meeting List*: Containing all requested meetings, including the meeting ID, organiser, and meeting members.

After having introduced the concept and its implementation, we will conclude this article in the next section with a summary and an outlook.

6 Summary and Outlook

We introduced a novel concept and its implementation as the TeamMeetingArranger designed to optimise team productivity by scheduling meetings based on real-time cognitive load assessments. By minimising interruptions during periods of high cognitive load, our approach aims to reduce the negative impact of interruptions on individual and team performance, especially in but not limited to collaborative work settings like open-plan offices. This system facilitates smoother transitions between focus and collaboration, ultimately enhancing productivity and well-being in the workplace.

The key components of the TeamMeetingArranger include a centralised coordinator and client-side applications. Through the continuous monitoring of cognitive load using eye-tracking technology, particularly the measurement of pupil diameter, the system identifies optimal meeting times. Once all team members' cognitive load levels drop below a dynamic threshold, the system notifies participants and initiates the meeting. This approach ensures meetings to only occur at moments of low cognitive load, helping to preserve focus and reduce task-switching costs.

Our current implementation demonstrates the feasibility of this approach, incorporating eye-tracking and a web-based client interface for meeting requests and management. The system adapts seamlessly to a typical workday, offering flexible configurations for dynamic cognitive load thresholds and meeting availability. The current implementation works in constant conditions—especially light conditions. Changing circumstances (e.g., stark changing to the light) can lead to noise in the eye tracking data and challenge the current implementation.

In the future, we aim to extend the system with the possibility of time-slot predictions for meetings based on historical data and to integrate user feedback to fine-tune decisions by not only relying a single feature, i.e., the cognitive load of users. Moreover, we plan a systematic user study of the TeamMeetingArranger to evaluate the long-term impact on productivity and workplace satisfaction.

Acknowledgments. We thank the members of the Cooperative Media Lab at the University of Bamberg.

References

1. Tom Gross, Michael von Kalben. 2023. A Literature Review on Positive and Negative Effects of Interruptions and Implications for Design. *Proc. INTERACT 2023*. https://doi.org/10.1007/978-3-031-42293-5_38
2. Florian Schaule, Jan Ole Johanssen, Bernd Bruegge, Vivian Loftness. 2018. Employing consumer wearables to detect office workers' cognitive load for interruption management. *Proc. ACM HCI*, 2, 1 (2018), 1-20. https://doi.org/10.1145/3191764
3. Tony Gillie, Donald Broadbent. 1989. What makes interruptions disruptive? A study of length, similarity, and complexity. *Psychological research*, 50, 4 (1989), 243-250. https://doi.org/10.1007/BF00309260
4. Shamsi T Iqbal, Brian P Bailey. 2005. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. *Proc. CHI 2005*. https://doi.org/10.1145/1056808.1056948
5. Maria K. Eckstein, Belén Guerra-Carrillo, Alison T. Miller Singley, Silvia A. Bunge. 2017. Beyond eye gaze. *Developmental Cognitive Neuroscience*, 25 (2017), 69-91. https://doi.org/10.1016/j.dcn.2016.11.001
6. Moritz Maleck, Tom Gross. 2024. CoLoTiMa: A Cognitive-Load Based Time Management Tool. *Proc. M&C 2024*. https://doi.org/10.1145/3670653.3677485
7. Ioanna Katidioti, Jelmer P Borst, Douwe J Bierens de Haan, Tamara Pepping, Marieke K van Vugt, Niels A Taatgen. 2016. Interrupted by your pupil. *International Journal of Human–Computer Interaction*, 32, 10 (2016), 791-801. https://doi.org/10.1080/10447318.2016.1198525

8. J Gregory Trafton, Christopher A Monk. 2007. Task interruptions. *Reviews of human factors and ergonomics*, 3, 1 (2007), 111-126. https://doi.org/10.1518/155723408X299852
9. Tom Gross. 2013. Supporting Effortless Coordination: 25 Years of Awareness Research. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 22, 4-6 (Aug.-Dec. 2013), 425-474. https://doi.org/10.1007/s10606-013-9190-x
10. Tom Gross, Anna-Lena Mueller. 2021. *A Formal Model of Availability to Reduce Cross-Domain Interruptions*. Proc. IHIET 2021. https://doi.org/10.1007/978-3-030-85540-6_3
11. Tom Gross, Anna-Lena Mueller. *NotificationManager: Personal Boundary Management on Mobile Devices*. Proc. INTERACT 2021. https://doi.org/10.1007/978-3-030-85610-6_15
12. Dorota Węziak-Białowolska, Zhao Dong, Eileen McNeely. 2018. Turning the mirror on the architects: A study of the open-plan office and work behaviors at an architectural company. *Frontiers in Psychology*, 9 (2018), 2178. https://doi.org/10.3389/fpsyg.2018.02178
13. David J Faulds, PS Raju. 2021. The work-from-home trend: An interview with Brian Kropp. *Business Horizons*, 64, 1 (2021), 29. https://doi.org/10.1016/j.bushor.2020.10.005
14. Godfred Anakpo, Zanele Nqwayibana, Syden Mishi. 2023. The impact of work-from-home on employee performance and productivity: a systematic review. *Sustainability*, 15, 5 (2023), 4529. https://doi.org/10.3390/su15054529
15. Michael Gibbs, Friederike Mengel, Christoph Siemroth. 2023. Work from home and productivity. *Journal of Political Economy Microeconomics*, 1, 1 (2023), 7-41. https://doi.org/10.1086/721803
16. Tanja Van der Lippe, Zoltán Lippényi. 2020. Co-workers working from home and individual and team performance. *New technology, work and employment*, 35, 1 (2020), 60-79. https://doi.org/10.1111/ntwe.12153
17. Harshad Puranik, Joel Koopman, Heather C Vough. 2021. Excuse me, do you have a minute. *Journal of Applied Psychology*, 106, 12 (2021), 1867. https://doi.org/10.1037/apl0000875
18. Hanna Kalmanovich-Cohen, Steven J Stanton. 2024. Leveraging quiet: The power of choosing your workspace. *Industrial and Organizational Psychology*, 17, 3 (2024), 371-374. https://doi.org/10.1017/iop.2024.23
19. Sophie Leroy, Aaron M Schmidt, Nora Madjar. 2021. Working from home during COVID-19: A study of the interruption landscape. *Journal of Applied Psychology*, 106, 10 (2021), 1448. https://doi.org/10.1037/apl0000972
20. Johannes Zagermann, Ulrike Pfeil, Harald Reiterer. 2016. Measuring Cognitive Load using Eye Tracking Technology in Visual Computing. *Proceedings of BELIV 2016*. ACM Press. https://doi.org/10.1145/2993901.2993908
21. Johannes Zagermann, Ulrike Pfeil, Harald Reiterer. 2018. Studying eye movements as a basis for measuring cognitive load. *Proc. Extended Abstracts of the CHI 2018*. https://doi.org/10.1145/3170427.3188628
22. Bastian Pfleging, Drea K Fekety, Albrecht Schmidt, Andrew L Kun. 2016. A model relating pupil diameter to mental workload and lighting conditions. *Proc. CHI 2016*. https://doi.org/10.1145/2858036.2858117
23. Daphne P. Dionisio, Eric Granholm, William A. Hillix, William F. Perrine. 2001. Differentiation of deception using pupillary responses as an index of cognitive processing. *Psychophysiology*, 38, 2 (2001), 205-211. https://doi.org/10.1111/1469-8986.3820205
24. Moritz Maleck, Tom Gross. 2023. AnswerTruthDetector: A Combined Cognitive Load Approach for Separating Truthful from Deceptive Answers in Computer-Administered Questionnaires. *i-com - Journal of Interactive Media*, 22, 3 (2023), 241-251. https://doi.org/10.1515/icom-2023-0023