Human Factors in Simulation, Software and Systems Engineering, Vol. 220, 2026, 49–59
https://doi.org/10.54941/ahfe1007693

Annoyance Modeling in Cooperative Personnel Scheduling

Christiane Attig[1], Johannes Varga[2], Tim Schrills[3], Tobias Rodemann[1], and Günther Raidl[2]

[1]Honda Research Institute Europe GmbH, 63067 Offenbach, Germany
[2]TU Wien, 1040 Vienna, Austria
[3]University of Lübeck, 23562 Lübeck, Germany

ABSTRACT

Cooperative algorithmic systems that depend on human input seldom model users' cognitive or affective states, even though these states can influence data input quality and, consequently, computational results. This paper demonstrates how annoyance – a prototypical user state elicited when systems repeatedly request input – can be modeled in a scheduling context and integrated into an algorithmic optimization process. We consider an industrial job scheduling scenario in which a cooperative scheduling system coordinates employee access to a shared machine. Rather than requesting all availability information upfront, the system iteratively improves an initial suboptimal schedule by querying users across multiple interaction rounds, subject to individual availability constraints and the goal of minimizing operational costs. Such repeated interactions can cause annoyance to accumulate, potentially resulting in careless responding or cooperation breakdown. To investigate the implications of modeling annoyance in this context, we simulated interactions between users and the scheduling system. Simulated users followed behavioral rules based on individual availability profiles, with annoyance building up across interaction rounds until an individual threshold was exceeded, beyond which users stopped responding truthfully. Results from 45,360 simulation runs showed that integrating annoyance modeling makes the algorithm more robust to parameter variations and yields better performance under suboptimal parameter choices than optimizing for cost reduction alone. These findings demonstrate that engineering psychology and algorithm design can mutually benefit one another: interactive scheduling offers a domain for applied user experience research, while engineering psychology provides algorithm designers with concepts to better account for users' willingness to cooperate with algorithmic systems.

Keywords: Job Scheduling, Annoyance, Algorithmic optimization, User experience

INTRODUCTION

Although algorithmic optimization increasingly affects human lives, human factors are often neglected in algorithm design. For instance, systems that depend on human input rarely model users' cognitive or affective states, even though these states can influence the quality of the data users provide and, consequently, the quality of computational results. Personnel scheduling is a prototypical example: As organizational complexity grows, automated scheduling systems are gaining importance (Bebien et al., 2025). Yet full

Received May 15, 2026; Revised June 20, 2026; Accepted July 6, 2026; Available online July 20, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

automation remains infeasible in this domain. Rather than solving the scheduling problem in one pass, cooperative scheduling systems iteratively improve an initial, suboptimal schedule by requesting availability information from users across several interaction rounds, subject to individual constraints and the goal of minimizing operational costs (Varga et al., 2024). The quality of the final schedule therefore directly depends on users' willingness to cooperate – to respond honestly and consistently across rounds. This makes personnel scheduling a fundamentally cooperative interaction between humans and intelligent systems, in which user states matter both for user experience and for algorithmic performance.

A key yet underexplored user state in this context is annoyance. When scheduling systems repeatedly request availability information from users, particularly at inconvenient times or across multiple interaction rounds, they can impose a cognitive and emotional burden (Sonnentag et al., 2018). We argue that this burden can cause annoyance to accumulate over time, eventually leading to careless responding or cooperation breakdown – and consequently to poorer data quality and suboptimal schedules (Ward & Meade, 2023). Integrating a model of annoyance into the scheduling algorithm itself thus offers a principled way to improve not only user experience, but algorithmic robustness.

In this work, we present a conceptual and technical implementation of an annoyance-aware scheduling algorithm in an industrial job scheduling scenario, in which a cooperative system coordinates employee access to a shared machine. We model annoyance as building up with each interaction round, with individual thresholds beyond which users stop responding truthfully, and integrate this model into an existing optimization framework (Varga et al., 2024). Results from simulation studies show that accounting for annoyance alongside operational costs improves algorithmic performance and robustness under suboptimal parameter choices. More broadly, our results demonstrate that the exchange between engineering psychology and algorithm design is mutually beneficial: interactive scheduling offers engineering psychology a domain for studying user states in context, while engineering psychology provides algorithm designers with concepts to better account for the human cooperation their systems depend on.

AUTOMATED SCHEDULING FROM A PSYCHOLOGICAL PERSPECTIVE

Automated scheduling can be understood as a resource allocation task in which human-centered systems assist or replace humans in assigning shifts, optimizing timetables, or managing complex resource distributions (Howard et al., 2020). The scheduling process generally comprises four stages, analogue to the stages of human information processing (Wickens & Carswell, 2021): information acquisition (e.g., gathering staff availability), analysis (e.g., verifying compliance with constraints), decision (e.g., generating an optimized schedule), and execution (e.g., finalizing assignments). Automation can take over parts of this process, but in cooperative scheduling systems, the information acquisition stage remains inherently dependent on human input. Recently, cooperative scheduling systems have been proposed that iteratively improve an initial suboptimal schedule by querying users across multiple interaction rounds (Varga et al., 2024), intended to reduce the initial burden on users while progressively refining the schedule toward operational goals.

This cooperative design, however, introduces a paradox (Bainbridge, 1983): The very mechanism intended to reduce user burden can itself become burdensome. Each interaction round places demands on users' time and attention. When requests are frequent, poorly timed, or perceived as yielding little improvement, they can be perceived as interruptions (Sonnentag et al., 2018), diverting attention from ongoing tasks and requiring repeated reallocation of cognitive resources (Lee et al., 2018). Designers therefore face a dilemma: Too little user input undermines scheduling accuracy, while too many queries risk alienating users who perceive the system as intrusive (Hertzum and Hornbæk, 2023). Research on interruptions consistently shows that enforced, excessive, or inopportune interactions impair concentration, heighten stress, and degrade task performance (Russell et al., 2021; Stangl & Riedl, 2023). In the context of cooperative scheduling, we argue that repeated system requests can cause annoyance to accumulate in users over interaction rounds. This is supported by findings from psychological questionnaire development research, which show that willingness to respond declines with increasing numbers of requests, particularly under stress (Bowling et al., 2021; Ashley & Shaughnessy, 2023). While the scheduling context differs from survey-taking, the underlying dynamic is analogous: Each additional request marginally increases the probability that a user's tolerance threshold is exceeded. Once that threshold is crossed, users may begin responding carelessly (i.e., providing inaccurate or strategically distorted availability information) or disengage from the interaction entirely (O'Brien & Toms, 2008). Either outcome degrades the data quality on which the algorithm depends, potentially resulting in suboptimal schedules despite the system's computational sophistication. This suggests that a reactive approach to annoyance (i.e., detecting and responding to it after it manifests) is insufficient. A more effective strategy is to embed sensitivity to annoyance directly into the scheduling algorithm, enabling it to proactively balance operational cost optimization against the cumulative interaction burden placed on users (Borst et al., 2022).

APPLICATION OF PSYCHOLOGICAL THEORY TO DESIGN AGENT-BASED EXPERIMENTS

To evaluate how scheduling systems can minimize unnecessary user interactions while maintaining scheduling quality, we developed and compared two algorithmic strategies: one that simply limits the number of availability requests users receive, and one that explicitly models and minimizes user annoyance. The interaction between the simulated user and the scheduling system, including how annoyance develops in response to system queries and how the system adapts its behavior accordingly, is illustrated in Figure 1.

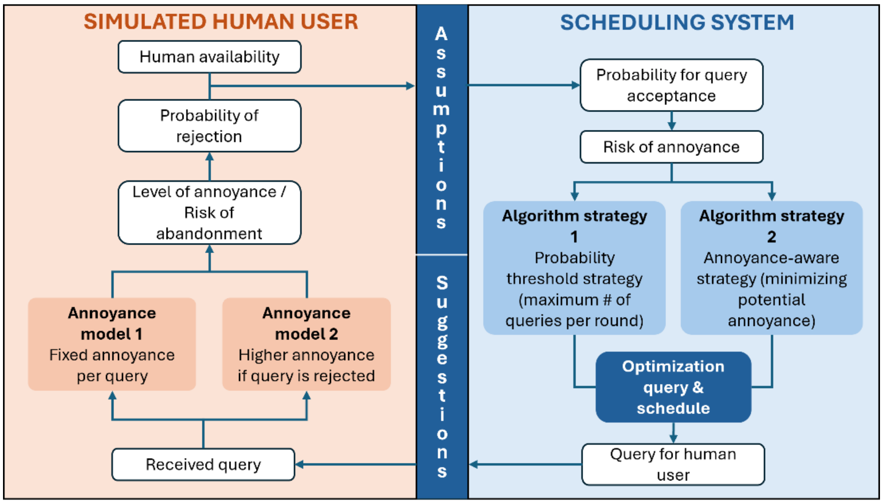

Figure 1: Depiction of two interacting simulations, with the left side depicting a model of how annoyance builds up and affects the user, and the right side depicting a model of the scheduler system.

Considered Scheduling Scenario and Experimental Setting

We considered a basic cooperative personnel scheduling scenario typical for industrial environments, where multiple employees must be given access to shared machinery over a fixed time horizon. Each of the twelve employees in our simulated scenario seeks to run four distinct jobs, all of which require uninterrupted, exclusive access to a machine and the user's presence to operate it. Notably, running a job incurs time-dependent operational costs (e.g., electricity).

To assist with this complex coordination task, an automated scheduling system (i.e., a central decision-making agent based on an algorithm) constructs schedules by iteratively interacting with employees to gather availability information. First, employees communicate their jobs and preferred timeslots and then the system prompts them with alternative job placements, which the employees can either accept or reject.

This exchange is repeated across multiple rounds, with the system refining its understanding of user constraints and preferences based on accumulated feedback. Importantly, each interaction constitutes a potential point of cognitive effort for users, requiring them to assess proposals, manage priorities, and potentially reconcile trade-offs between individual convenience and organizational efficiency. After the final interaction round, the system terminates with the last computed schedule that considers the whole collected data. While the formal optimization details are beyond the scope of this work, a full specification of the algorithmic framework can be found in (Varga et al., 2024).

User Simulation and Operationalization of Psychological Constructs

User availability model. We simulated the user-algorithm interaction (i.e., responses to system queries) of twelve individual employees based on realistic work patterns. Each simulated user had their own availability profile, which we derived from the Dutch Time Use Survey (Cultureel Planbureau, 2005) by extracting presence times at the workplace. For their initial job preferences, each user selects time slots that fit within their individual availability. In the beginning, these initial preferences are the only information the scheduling system has available about the users (i.e., the system does not have access to the users' availability profiles). When the scheduling system later proposes alternative time slots through queries, users accept a new time only if it aligns with their availability as defined in their respective availability profile.

Annoyance model. To model annoyance, we differentiate between the emergence of annoyance and its behavioral consequences. For modeling the emergence (i.e., accumulation) of annoyance, two alternative approaches were developed, implemented, and compared. The first model reflects the assumption that each additional request for information imposes a roughly equal cognitive or affective cost on the user and models the annoyance linearly in the total number of queries a user receives:

$$\text{Annoyance} := \#\text{Queries}$$

The second model introduces a more nuanced distinction by assigning different weights to accepted and rejected queries, which reflect the assumption that not all interactions are equally taxing:

$$\text{Annoyance} := 0.5 \cdot \#\text{Accepted} + 2 \cdot \#\text{Rejected}$$

Compared to an accepted query, a rejected query might represent a greater interruption or annoyance, since it can be perceived as a sign that the system has misunderstood the user or is not taking their constraints seriously. This can lead to greater negative affect, particularly when users feel that they have already communicated relevant information and yet are prompted to negotiate further (i.e., the need to reject queries might signal low system cooperativity; Attig et al., 2024).

Second, we understand a user's annoyance as a scalar value in dependence of the answered queries and introduce an annoyance tolerance: After exceeding an annoyance threshold, simulated users begin rejecting all further requests, regardless of availability. This behavior models excessive interactional burden leading to disengagement and careless responding (Ward & Meade, 2023). To account for individual differences, each user gets an individual annoyance threshold, randomly drawn from a range of values (Meys & Sanderson, 2013).

Algorithm Strategies

Probability threshold strategy. To ask just enough of the right questions, the scheduling system works in multiple steps. First, it generates a query candidate for each job and for many possible alternative time slots, such as: "Could you perform this job on Tuesday at 2 PM instead?". Based on what the system knows about a user's availability from previous responses, it estimates the likelihood that the user will accept each suggestion, applies an acceptance threshold (p_{lim}, from the interval 0 to 1), and only keeps queries that meet this threshold. From the remaining candidates, the system selects a small number of the most helpful queries – for instance, $b = 12$

per round. The value of *b* is fixed in the beginning of the simulation and does not change. The selection prioritizes queries that are expected to improve the schedule quality, considering factors such as electricity costs, and is done by computing an optimized schedule with the help of Integer Linear Programming–an advanced algorithmic optimization technique–see (Varga et al., 2024) for more details on how queries are computed. After users respond, the system updates its knowledge of their availabilities and generates a revised preliminary schedule. This process is repeated over *B* rounds.

Annoyance-aware strategy. We modify the above basic approach to account for users' annoyance when selecting queries. In this version, the system aims to balance two factors: reducing the operational costs of the schedule and minimizing user annoyance. Instead of simply maximizing the cost reduction, the system now tries to maximize:

$$\textit{Cost reduction} - \gamma \cdot \textit{Expected Annoyance}.$$

The parameter $\gamma \geq 0$ is used to adjust how much the system values user annoyance compared to cost reduction. A higher value of γ means that the system will prioritize minimizing annoyance more heavily, whereas lower values will favor cost reduction. As γ determines how many queries are selected, the former limit on the number of queries b no longer applies. The expected result is a scheduling system that, by adjusting γ, can better balance the need for efficient scheduling with the psychological cost of too many interactions with the users.

Table 1: Overview of algorithm approaches and parameters.

	Annoyance Modeling	
	Linear Function	**Reply-Based**
General idea	Annoyance level increases with number of queries per round in a linear fashion, regardless of response to query (acceptance or rejection).	Annoyance level increases with number of queries per round, but with different weights for acceptance and rejection (rejected queries elicit stronger annoyance).
Behavioral response to annoyance	Users begin rejecting all further requests, regardless of availability, if individual annoyance threshold is passed.	
	Algorithm Strategy	
	Probability threshold	**Annoyance-aware**
General idea	The system limits the number of queries to a fixed amount while still gathering enough information to enable effective scheduling. Requests to change time slots are made based on cost considerations and the likelihood that the user will accept them, as estimated by the system using its partial knowledge of the user's availability.	The number of system queries is dynamically limited to balance cost efficiency and minimize user annoyance. Requests to change time slots are made based on cost considerations and the likelihood that the user will accept them, as estimated by the system using its partial knowledge of the user's availability.

(Continued)

Table 1: Continued.

	Algorithm Strategy	
	Probability Threshold	**Annoyance-Aware**
Parameter description, range, and settings for simulations	B = Number of interaction rounds ($B \geq 1$)	
	$B = 1, 2, \ldots, 10$	
	p_{lim} = Acceptance probability threshold, that is, set likelihood threshold that user will accept a query ($0\% \leq p_{\text{lim}} \leq 100\%$)	
	$p_{\text{lim}} = 10\%, 20\%, \ldots, 90\%$	
	b = Upper limit for the number of queries ($b \geq 0$)	–
	$b = 8, 12, 16$	
		γ = Weight of the annoyance ($\gamma \geq 0$)
	–	$\gamma = 0.0, 1.0, 1.5, 2.0, 3.0, 5.0, 7.0, 10.0, 15.0, 20.0, 30.0$

QUANTITATIVE EVALUATIONS

To test the algorithmic approaches, we utilized simulations. In the following, we present simulation settings and quantitative results of the simulations with the two algorithmic approaches and artificial users.

Simulation Setup

We consider a scenario representative of typical industrial environments, where a primary objective is reducing operational costs, made up of time-dependent electricity costs and expected future expenses from neglecting a job altogether. These operational considerations must be carefully balanced against employee annoyance, which directly impacts employee satisfaction and productivity. To better understand how different scheduling systems (i.e., algorithmic approaches) might affect user annoyance and scheduling costs, we conducted simulations of the interaction between the scheduling system and users, systematically varying parameters of the scheduling algorithm under different assumptions about user availabilities and behavior (i.e., user models). The simulations are based on realistic job scheduling scenarios with two shared machines, twelve users, and four jobs per user over a five-day planning period. For more reliable results, we averaged outcomes across nine different task configurations[1]. We used the implementation of (Varga et al., 2024) realized in the programming language Julia 1.10.5 (https://julialang.org) and applied the Gurobi optimization library (https://www.gurobi.com) to compute efficient schedules.

To capture individual differences in sensitivity to interaction load, we modeled two types of users: one group with lower annoyance tolerance (annoyance threshold values ranging from 3 to 6), and one group with higher

[1]Instances 1 to 9 from https://www.ac.tuwien.ac.at/research/problem-instances/#ijsp

tolerance (annoyance threshold values ranging from 5 to 8). Each group was further tested under the two different annoyance models: the simple linear model, where the annoyance value is the number of queries, and the more nuanced reply-based model, where accepted and rejected queries contribute differently to annoyance. Each of these four user types was tested under the two scheduling approaches described above: the probability threshold-based approach and the annoyance-aware approach. Table 1 outlines the parameter settings explored. Each unique parameter combination was used in the simulation. For each combination of parameter values in one of the approaches, each user profile and each instance, one simulation run was performed, resulting in 45,360 runs in total.

Evaluation Results

We examined the influence of the acceptance probability threshold p_{lim}; remember that it determines how confident the system must be that a user will accept a query before considering it further. Figures 2 and 3 show for different values of p_{lim} the lowest costs obtained over all tested parameters of the other parameters. Each curve in the four diagrams has been created for one combination of annoyance model (linear vs. reply-based), user group (lower vs. higher annoyance tolerance), and algorithmic approach (probability threshold vs. annoyance-aware); to avoid overloading the plots, we show a selection of them.

First, we have a look at the differences between the resulting curves of the linear and the reply-based annoyance model (see Figure 2). The simulations revealed that with the linear annoyance model, the best outcomes were achieved with a probability threshold p_{lim} of 0.7, whereas in the reply-based user model, a higher threshold of 0.8 worked better, irrespective of the algorithmic approach. This underscores the necessity of accurately modeling user behavior to make informed choices when adjusting algorithm parameters in practical deployments.

Now, we compare the curves of both approaches (see Figure 3). While a suboptimal choice of the parameter p_{lim} incurs additional costs, the increase in these costs is notably smaller when employing the annoyance-aware approach. Compared to the threshold approach, costs are reduced by 10.4% and 8.6% for the lower and higher tolerance groups, respectively, when using $p_{\text{lim}} = 0.4$. This shows that the approach that explicitly incorporates user-centric considerations exhibits better robustness to suboptimal parameter settings. In a practical deployment scenario, suboptimal parameter settings need to be considered, because parameter choices would be based on a simulation with the algorithmic approach and a user model. Since user models cannot capture all nuances of actual user behavior, discrepancies between predicted and observed behaviors are inevitable, underlining the importance of robust algorithmic approaches. This is particularly valuable since optimal parameters are sensitive to variations in user behavior.

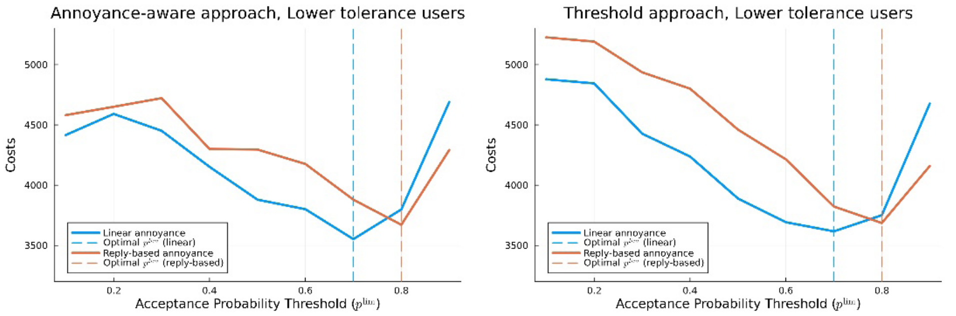

Figure 2: Final costs depending on the acceptance probability threshold plim when optimally choosing the other scheduler parameters. Compares the two user models (linear and reply-based annoyance) for both approaches. When annoyance is modeled reply-based higher threshold values for p_{lim} need to be chosen in the scheduler to minimize costs.

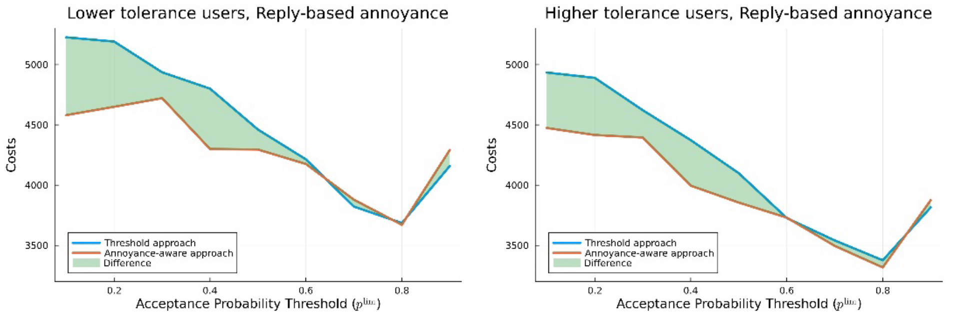

Figure 3: Final costs depending on the acceptance probability threshold p_{lim} when optimally choosing the other scheduler parameters. Compares the two approaches (threshold and annoyance-aware) for the two groups of users. The annoyance aware approach produces much lower costs when a sub-optimal value is selected for p_{lim}.

DISCUSSION

Automated scheduling systems are becoming increasingly important in complex organizational settings, yet human factors are often overlooked in algorithm development, even though they directly affect whether users provide the accurate information these systems depend on. In this paper, we presented an algorithmic approach that incorporates theoretically grounded assumptions about user annoyance: a state assumed to build up with repeated system queries and, once an individual threshold is exceeded, to result in non-cooperation. Evaluation results across various model specifications demonstrated that the annoyance-aware approach most effectively balanced informational richness with operational costs and proved more robust to suboptimal parameter choices than a cost-only strategy.

From an algorithm development perspective, our results illustrate both how and why human factors should be integrated into cooperative scheduling algorithms. Psychological parameters such as individual annoyance thresholds demonstrably affect internal system parameters, meaning that optimal human-algorithm integration is not merely a theoretical question for engineering psychology but a practical challenge for algorithm developers. Notably, our annoyance model is interpretable, traceable, and lightweight, which are properties that make it a realistic candidate for integration into existing optimization frameworks.

From an engineering psychology perspective, the presented work demonstrates that even simple, psychologically informed user modeling can meaningfully improve algorithmic performance. However, simulation can only be a first step. User studies assessing subjective experience, perceived system cooperativity, and behavioral response patterns are essential to validate our assumptions and deepen understanding of human-algorithm integration. Beyond avoiding annoyance, future work might also explore how interactions can actively support positive user experience, for instance, by addressing users' basic psychological needs for autonomy, competence, and relatedness (Ryan & Deci, 2000), thus moving from the prevention of negative states toward the promotion of genuinely fulfilling engagement with scheduling systems.

ACKNOWLEDGMENT

J. Varga acknowledges financial support from Honda Research Institute Europe.

REFERENCES

Ashley, M. and Shaughnessy, K. (2023) 'Predicting insufficient effort responding: The relation between negative thoughts, emotions, and online survey responses', *Canadian Journal of Behavioural Science*, 55(3), pp. 198–209.

Attig, C., Wollstadt, P., Schrills, T., Franke, T. and Wiebel-Herboth, C.B. (2024) 'More than task performance: Developing new criteria for successful human-AI teaming using the cooperative card game Hanabi', *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. ACM, pp. 1–11.

Bainbridge, L. (1983) 'Ironies of automation', *Automatica*, 19(6), pp. 775–779.

Bebien, V., Bellenguez, O., Coppin, G., Ma-Wyatt, A. and Stephens, R. (2025) 'Ethical decision-making in human-automation collaboration: A case study of the nurse rostering problem', *AI and Ethics*, 5(2), pp. 1163–1175.

Borst, J., Bulling, A., Gonzalez, C. and Rußwinkel, N. (2022) 'Anticipatory human-machine interaction (Dagstuhl Seminar 22202)', *Dagstuhl Reports*, 12(5), pp. 131–169.

Bowling, N.A., Gibson, A.M., Houpt, J.W. and Brower, C.K. (2021) 'Will the questions ever end? Person-level increases in careless responding during questionnaire completion', *Organizational Research Methods*, 24(4), pp. 718–738.

Cultureel Planbureau, S. (2005) *Tijdsbestedingsonderzoek 2005 - TBO 2005*. DANS Data Station Social Sciences and Humanities.

Hertzum, M. and Hornbæk, K. (2023) 'Frustration: Still a common user experience', *ACM Transactions on Computer-Human Interaction*, 30(3), pp. 1–26.

Howard, F.M., Gao, C.A. and Sankey, C. (2020) 'Implementation of an automated scheduling tool improves schedule quality and resident satisfaction', *PLOS ONE*, 15(8), article e0236952.

Lee, B.C., Chung, K. and Kim, S.-H. (2018) 'Interruption cost evaluation by cognitive workload and task performance in interruption coordination modes for human–computer interaction tasks', *Applied Sciences*, 8(10), article 1780.

Meys, H.L. and Sanderson, P.M. (2013) 'The effect of individual differences on how people handle interruptions', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 57(1), pp. 868–872.

O'Brien, H.L. and Toms, E.G. (2008) 'What is user engagement? A conceptual framework for defining user engagement with technology', *Journal of the American Society for Information Science and Technology*, 59(6), pp. 938–955.

Russell, E., Jackson, T. and Banks, A. (2021) 'Classifying computer-mediated communication (CMC) interruptions at work using control as a key delineator', *Behaviour & Information Technology*, 40(2), pp. 191–205.

Ryan, R.M. and Deci, E.L. (2000) 'Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being', *American Psychologist*, 55(1), pp. 68–78.

Sonnentag, S., Reinecke, L., Mata, J. and Vorderer, P. (2018) 'Feeling interrupted—being responsive: How online messages relate to affect at work', *Journal of Organizational Behavior*, 39(3), pp. 369–383.

Stangl, F.J. and Riedl, R. (2023) 'Interruptions in the workplace: An exploratory study among digital business professionals', in F. Nah, and K. Siau (eds.), *HCI in business, government and organizations*. Springer, pp. 400–422.

Varga, J., Raidl, G.R., Rönnberg, E. and Rodemann, T. (2024) 'Scheduling jobs using queries to interactively learn human availability times', *Computers & Operations Research*, 167, article 106648.

Ward, M.K. and Meade, A.W. (2023) 'Dealing with careless responding in survey data: Prevention, identification, and recommended best practices', *Annual Review of Psychology*, 74(1), pp. 577–596.

Wickens, C.D. and Carswell, C.M. (2021) 'Information processing', in G. Salvendy and W. Karwowski (eds.), *Handbook of human factors and ergonomics*. John Wiley & Sons, pp. 114–158.