Implicit Interactions in Proactive Systems: Evaluation Challenges and Adaptations for Nielsen's Heuristics

Alba Bisante[*,†], Alan Dix[*,†], Emanuele Panizzi[*,†] and Stefano Zeppieri[*,†]

[1] *Department of Computer Science, Sapienza University of Rome, Rome, Italy*
[2] *Computational Foundry, Swansea University, Wales, United Kingdom*
[1] *Department of Computer Science, Sapienza University of Rome, Rome, Italy*
[4] *Department of Computer Science, Sapienza University of Rome, Rome, Italy*

Abstract

This paper examines the challenges of applying established evaluation standards, such as Nielsen's heuristics, to the assessment of implicit interactions in proactive systems. By analyzing the spectrum of user intention in implicit interactions, this work highlights limitations in current usability approaches and explores potential adaptations to address these gaps. This work contributes to advancing evaluation practices for a new generation of intelligent, context-aware systems.

Keywords

proactive systems, implicit interaction, heuristic evaluation, HCI, AI

1. Introduction

With the growing interest in AI-based technologies and their increasing application in everyday systems, we are witnessing a noticeable shift from traditional interaction paradigms to more seamless and natural forms of interaction [1]. As users become more accustomed to smart, context-aware systems, expectations are rising for technology to anticipate and respond to needs with minimal direct input [2]. Proactive agents, powered by AI and based on implicit interaction, interpret user actions or contextual cues without requiring explicit commands, delivering useful features and services automatically, while offering a more intuitive and adaptive experience [3].

Such a massive shift towards the adoption of proactive agents poses new challenges for the design and for evaluating system usability. Established HCI evaluation methods, which are largely based on explicit user interaction, may not fully capture the complexity of implicit systems. Specifically, existing usability frameworks, including Nielsen's heuristics [4] and broader HCI principles such as learnability, flexibility, and robustness, may not directly apply to systems that function in the background and respond to unspoken cues.

This works aims to address the challenges of evaluating implicit interactions in proactive systems, recognizing the need for specific tools and criteria that align with the designer's intent to create natural and seamless experiences. A central focus is discussing the applicability of established Nielsen's evaluation heuristics, outlining how this framework can be adapted to evaluate implicit interactions in proactive systems.

The paper is organized as follows. First, section 2 provides an overview of relevant literature. Section 3 discusses the user's spectrum of intention in implicit interaction with proactive agents, with a focus on the relationship between implicit input and its resulting effects. In section 4 we address challenges

Joint Proceedings of the ACM IUI Workshops 2025, March 24-27, 2025, Cagliari, Italy

[*]Corresponding author.
[†]These authors contributed equally.
bisante@di.uniroma1.it (A. Bisante); alan@hcibook.com (A. Dix); panizzi@di.uniroma1.it (E. Panizzi); zeppieri@di.uniroma1.it (S. Zeppieri)
https://alandix.com/ (A. Dix); https://stefanozeppieri.com/ (S. Zeppieri)
0000-0002-5996-4221 (A. Bisante); 0000-0002-7442-8451 (A. Dix); 0000-0002-7442-8451 (E. Panizzi); 0000-0001-8392-2251 (S. Zeppieri)

posed by the application of the established Nielsen's evaluation heuristics. Finally, section 5 discusses and draws conclusions based on our findings.

2. Related Works

Proactive systems are intelligent technologies designed to autonomously anticipate user needs and act without explicit commands, relying on contextual data, historical behavior, and reasoning mechanisms [5, 1, 6]. A defining feature of these systems is implicit interaction, where actions are triggered by subtle cues such as sensor data or user activity, rather than direct input [2, 5].

Schmidt's early work [7] laid the foundation for implicit interaction by exploring how systems could infer user intentions without explicit input, emphasizing context-awareness and passive sensing. Building on this, Schmidt and Herrmann [8] introduced the concept of intervention interfaces, allowing users to retain control in highly automated systems, a principle closely related to balancing autonomy and user involvement in implicit interaction design.

Context-awareness has been extensively studied in HCI, with Cockton and Gram [9] highlighting the evolution of usability to accommodate non-traditional input in technologies like smart homes. Similarly, Dix's spectrum of intention [10] categorizes user interactions from explicit to incidental, providing a framework for understanding low-intention interactions and their seamless integration into users' primary tasks. Serim and Jacucci [11] extended this discussion, identifying key types of implicit interaction and emphasizing the need for systems to avoid making users feel manipulated or disempowered.

Proactive systems leveraging implicit interaction have been explored across various domains. For instance, conversational agents utilize cues such as conversational history and user preferences to tailor responses [12, 3]. Similarly, sensing through smartphone inertial sensors has proven effective for understanding diverse contexts; for example, in earlier studies we addressed earthquake detection [13] and the identification of driving-related behaviors [14, 15]. However, challenges remain in areas such as transparency, usability, and trust, as users often find it difficult to comprehend or anticipate system behavior [16].

Recent work has focused on evaluation frameworks for implicit systems. Serim and Jacucci [11] emphasized the importance of distinguishing between helpful and intrusive interactions in ubiquitous computing environments. Mueller et al. [17] and Bennett et al. [18] both explored how systems can preserve user autonomy while seamlessly integrating with natural behaviors. Additionally, Bisante et al. [19] highlighted the importance of error detection and repair in AI-based implicit systems to prevent interaction breakdowns.

Although established usability frameworks like Nielsen's heuristics [4] provide a foundation for evaluating explicit interactions, their application to implicit systems remains underexplored. Amershi et al. [20] and Helms and Brown [21] provide relevant insights for designing human-AI interactions and sensing-based systems, but further adaptation is required to address the unique challenges of implicit interaction.

Building on these works, this paper discusses how to apply Nielsen's heuristics to evaluate implicit interactions in proactive systems, advancing evaluation practices for intelligent, context-aware systems.

3. Implicit interaction with Proactive Agents

Proactive agents are characterized by their ability to anticipate user intentions, relying on contextual information, acting autonomously, and operating without requiring explicit user interaction. Instead, these systems use implicit signals, such as sensor data, historical patterns, and contextual information, to make decisions. In this work, we focus on the challenges of evaluating the implicit interaction part [5, 1, 6].

As noted by Serim and Jacucci [11], the concept of implicit interaction has been defined in various ways in the literature. Our interpretation of implicit interaction builds on Dix's spectrum of intention

[10] [22], which frames implicit interaction on a continuum of sensor-based interactions, categorized by the degree of user intention behind the use of sensed inputs.

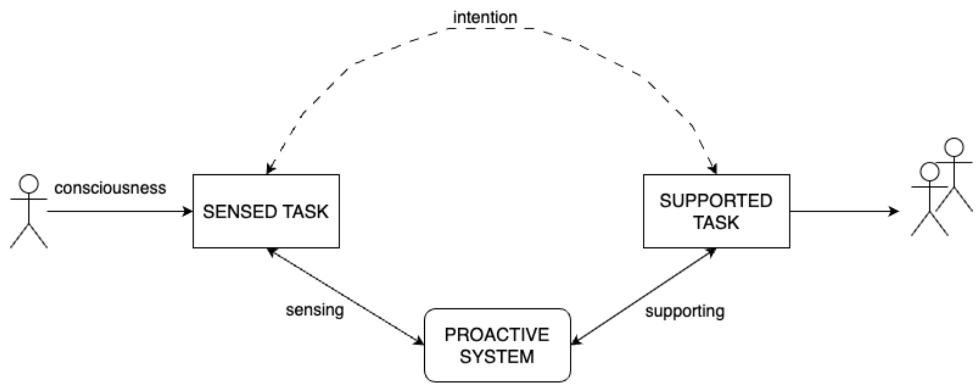

Figure 1: Human-system interaction for sensing and supporting tasks

At the high-intention end there are very explicit interactions, such as gesture-based systems, flicking a tablet to turn a page or annotation systems [23]. These are sometimes called 'implicit' as they recruit tacit knowledge and may become part of practiced use.

The opposite, low-intention end, (termed 'incidental' interactions) are situations where the user being sensed simply carries on their activities, with no explicit intention that the sensed data may be used by a proactive system (although they may know if they stop to think about it). It is this low-intention end of the spectrum that is the primary focus of this paper.

In these low-intention situations, it is possible to distinguish two user tasks, the sensed and supported tasks (called primary and secondary tasks in early literature):

- Sensed (primary) task: a task that the user performs regardless of the proactive system. The system senses what the user is doing to build knowledge to help in other tasks. For example, you watch a streamed film because you want to watch it, but the system builds a model of your own preferences, to help make suggestions for you or other users in the next choices.
- Supported (secondary) task: thanks to data sensed during the implicit interaction of the sensed tasks, the proactive system supports a 'future' user's task. As shown in fig. 2, the use of sensing data may happen in the same interaction episode (pink circle) or in subsequent episodes (orange/green circles). But also, several sensed tasks can help support a single or several supported tasks, done over time and/or done by different users. A supported task could be considered as non-proactive when the system gathers data during the sensed task and uses it to assist the user in a way that requires explicit input or triggers from the user. In this case, the system does not autonomously act but instead provides support based on user-initiated actions. In this work, we focus only on supported tasks that are proactive and based on implicit interaction.

4. Applicability of Nielsen's Heuristics

Nielsen's heuristics, traditionally applied to explicit, user-driven interfaces, are often more challenging to apply to proactive systems. In these systems, the interaction is shaped by context and user behavior rather than direct commands, and as a result, some heuristics may be less relevant or require reinterpretation.

In this section, we discuss which are the challenges of applying each of Nielsen's usability heuristics to proactive interfaces, referencing the spectrum of intentionality of the user and our focus within implicit interaction.

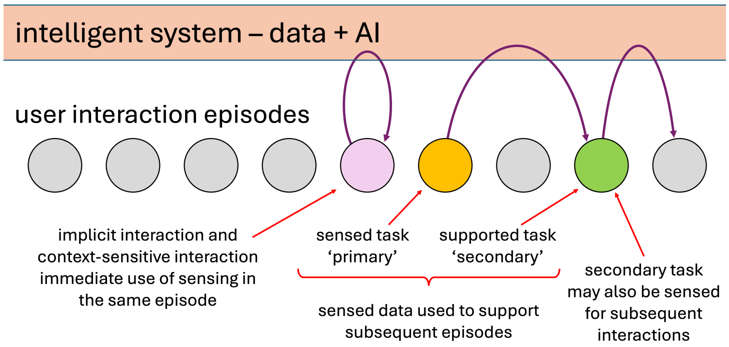

Figure 2: Sensed and supported tasks

4.1. Visibility of System Status

Visibility of system status helps users understand cause-and-effect relationships in interaction and predict the system's behavior. However, maintaining the visibility of system status is often challenging with proactive systems, as implicit data gathering during the sensed task inherently obscures the system's status, posing also potential privacy issues.

Users typically engage in the sensed task naturally, with the proactive system's sensing occurring in the background. This lack of explicit feedback may lead users to focus more on the outcomes of supported tasks than on the system status itself. Moreover, when the supported task is proactive, users may find it particularly difficult to understand why specific actions are being proposed.

For example, when a light automatically switches on, its status is immediately visible, offering clear feedback. In fact, in instances where the sensed and supported tasks overlap (the 'pink circle' scenario), the visibility of system status becomes more direct. However, in cases like air conditioning, where the effect (temperature change) is more gradual or less perceptible, the system status may be harder to detect.

In any case, if users can perceive the system's status - whether through feedback or other cues - it can help them form a mental model of how the system works. Thus, while the visibility of system status does not typically affect the sensed task directly (e.g., the user may not be consciously aware of background sensing), it can influence how the user engages with supported tasks.

4.2. Match Between the System and the Real World

This heuristic emphasizes that systems should use familiar concepts, language, and conventions to make interactions intuitive. In proactive systems, this applies especially on designing of the sensing process, hence the sensed tasks, to reflect real-world behaviors and expectations.

For example, users might expect lights to turn on based on their presence in a room, as this aligns with a real-world understanding of occupancy. However, if the system instead requires specific movement to activate the lights, it violates this heuristic by introducing an artificial constraint that does not match the user's mental model of presence and response. This mismatch can lead to confusion and frustration, as users may struggle to understand or predict the system's behavior.

The core challenge lies in designing proactive systems that not only interpret user behavior accurately but also translate this understanding into actions that resonate with the user's perception of how the real world operates.

4.3. User Control and Freedom

Users should ideally retain a degree of control over systems, including the ability to undo or cancel actions. However, in proactive systems, users often relinquish some level of control, as direct intervention would shift the interaction from implicit to explicit. Moreover, in many real-world implicit interaction scenarios, undoing implicit actions can be difficult or even impossible. For example, in systems like automatic toll collection at toll booths, once the user's vehicle passes through the toll gate, there is no way to undo the transaction or change the decision to take that particular route.

In proactive systems based on implicit interactions, control can only be exercised if users are aware of the system's operations and have a mental model of how the sensing mechanism works and how it supports secondary tasks. For instance, if users understand how a smart home system senses their movement to turn on lights, they may intentionally modify their behavior to manipulate the outcome. However, when they do this, the interaction moves from incidental to intentional—situating the user higher on the spectrum of intention.

It is important to recognize that users generally appreciate having control, and this can enhance the usability of a system. Even though implicit interactions reduce direct user control, the option to influence the system—once users develop awareness—can provide them with a sense of agency. This is not necessarily negative from a usability perspective. However, in the case of implicit interactions designed to operate in the incidental or low-intention spectrum, control is not inherently applicable. With respect to sensed and supported tasks, this heuristic mainly applies to the relationship between the sensed and supported tasks.

4.4. Consistency and Standards

When considering consistency within proactive systems, we propose an interpretation that relates to maintaining uniform behavior across similar elements within the same environment. For example, if a building has multiple automatic doors, all the doors should function in a consistent manner to meet user expectations.

Inconsistent behavior across similar systems — such as one door opening automatically and another requiring pressing a button — can disrupt the seamless experience that implicit interactions aim to provide. This heuristic applies mainly to the design of the relationship between sensed and supported tasks.

Regarding the concept of 'standards', we argue that it aligns with what feels most natural to the user. More specifically, there are two levels of standards to consider:

- **User expectations based on prior experience:** Users often anticipate certain behaviors from systems based on what they have encountered before. For example, users might not expect elevators to arrive automatically without needing to press a button, but they do expect automatic doors to open in modern environments, especially when there is no visible handle. These implicit standards are based on context and familiarity with similar interactions.
- **Naturalness as the standard:** The most natural interaction often becomes the "standard" for users. For example, if users were to encounter an elevator designed to operate automatically, they would naturally expect a simple action—such as approaching it—to trigger its functionality. In contrast, requiring an exaggerated or unnatural action to call the elevator would feel cumbersome and counterintuitive. The closer a system aligns with users' natural expectations, the smoother and more seamless the interaction becomes. Understandably, what is considered natural can vary among users and may depend on the complexity of the system or task. While acknowledging these variations, we propose this general definition as a starting point for discussing naturalness in interaction design.

4.5. Error Prevention

The HCI literature distinguishes between two types of errors: *slips*, which occur when users intend to perform the correct action but inadvertently perform the wrong one, and *mistakes*, which arise when users form incorrect intentions based on faulty mental models [24, 25].

In low-intention, implicit interaction systems, we argue that the risk of slips does not apply as the interaction is natural and not reliant on conscious input. However, mistakes can still happen, especially when users have an inaccurate understanding of how the system works. For example, a user may believe that their mere presence in a room will trigger the lights to turn on, when in fact, the system requires movement to activate the lights.

To prevent errors, particularly mistakes, Nielsen suggests several design considerations that remain relevant in proactive systems. Below, we discuss how each of these aspects applies to implicit interactions:

- **Minimize memory burdens:** Proactive systems should avoid requiring the user to recall prior actions or states. Requiring memory recall shifts the interaction from implicit to explicit, undermining the system's seamless nature. The design should ensure that the system reacts naturally to the current context without placing cognitive demands on the user. This aspect applies to sensed tasks.
- **Support a simple mental model:** A user's mental model of the system should be as simple and accurate as possible. In a proactive system, if users misunderstand how their actions trigger responses, they may form incorrect expectations and make mistakes. For instance, when the user mistakenly assumes that being inside a room activates the lights when the system actually senses motion detection. Clear system feedback or subtle cues can help form more accurate mental models. This aspect involves both sensed and supported tasks.
- **Warning users:** This should never be necessary for the sensed task, as the goal is to simply observe the user. If there is a need to warn users about problems, for example "please do not sit still for too long", the system has failed. It is the designer's responsibility to ensure that these problems do not occur. This said, no system is perfect, and there may be times when breaking the illusion of invisibility may be necessary. A good example of this is in fall-detection systems: at the point a fall has been detected, most will offer ways to warn the users that their posture (maybe simply having a yoga session on the floor) is going to be treated as an incident before the ambulance is called. However, the general rule is that these are rare exceptions.

Moreover, proactive, intelligent systems today are often powered by AI and are typically trained on limited datasets, relying on constrained sensor input to interpret the environment, including users' behaviors and intentions. This inherent limitation makes such systems prone to system errors, further underscoring the critical need for designers to prioritize robust error-handling mechanisms. Effective error handling should take into account not only user errors, but also is particularly crucial to prevent system errors from escalating into full system failures, ensuring the reliability and usability of these systems [19].

4.6. Recognition Rather than Recall

This heuristic does not usually apply to implicit proactive systems. If users need to recall specific actions or information, the interaction becomes explicit. In a low-intention implicit system, users should not need to remember prior inputs for the system to function correctly. In practice, there can be some middle ground, for example, if people modify their walking speed as they approach an automatic door, which is effectively a form of (implicit) recall, but the underlying assumption is that the basic implicit interaction functions well enough based on the users normal (non AI-assisted) behaviour and so does not require explicit means to provide recognition. Thus, when applicable, this heuristic applies to sensed tasks.

4.7. Flexibility and Efficiency of Use

The heuristic of flexibility and efficiency of use suggests that systems should allow users with different skill levels or preferences to interact with the system in ways that best suit their needs.

In implicit interactions, this flexibility can manifest through personalization across two dimensions:

- **Interaction personalization:** Different implicit interactions can trigger the same outcome based on the user. For instance, a light might turn on when one user sits down at a table, but for another user, it activates when they enter the room. This dimension applies to sensed tasks.

- **Content personalization:** Different outcomes can result from the same interaction depending on the user. For example, one user's presence may trigger only specific lights to turn on, while another user's presence activates all the lights. This dimension applies to supported tasks.

Customization in the traditional sense is less relevant in proactive systems, as it requires explicit interaction. Likewise, accelerators (shortcuts) are not applicable since implicit interactions are already designed to be the most efficient, natural responses.

In contrast, adaptations are common in proactive systems, for example, if the user frequently turns the lights to a dimmed setting after entering the room, the systems can adapt to default to that setting. However, the challenge is that normal adaptation should also be based on implicit cues and not on explicit user settings.

4.8. Aesthetic and Minimalist Design

Although aesthetics may not be the first consideration in implicit interactions, this heuristic still holds relevance. The design of proactive systems should prioritize simplicity and elegance, both in the selection of the sensed task and the support of the secondary.

4.9. Help Users Recognize, Diagnose, and Recover from Errors

This heuristic is less applicable in proactive systems. While feedback on the execution of a supported task can help users recognize when something has gone wrong (e.g., a door failing to open or a light not turning on), diagnosing and resolving such errors often requires explicit user intervention through traditional interfaces. This can be a challenge in systems relying on implicit interactions, which inherently reduce the user's direct control.

A common approach to designing for error recovery involves supporting undo functionalities. This can be achieved by providing contextual feedback that helps users understand and correct unintended actions. However, as discussed under *User Control and Freedom*, undoing actions in physical implicit interactions poses greater challenges compared to traditional digital systems. Furthermore, enabling undo functionality in these contexts often requires shifting the user toward a more deliberate, high-intention mode of interaction.

Consequently, this heuristic primarily applies to the "recognize" aspect. Any system failure should be clearly communicated to the user through explicit feedback mechanisms, ensuring the user can identify the issue and take corrective action. Challenges related to error detection and recovery in AI-based user interfaces have been explored in detail in earlier work [19].

4.10. Help and Documentation

This heuristic tends to be less relevant for proactive systems, especially in low-intention interactions where the system monitors users' natural behaviors and responds automatically. In these cases, if documentation is needed to explain how to interact with the system, it may indicate that the interaction is no longer truly intuitive. Ideally, proactive systems should be designed to operate in the background, with users engaging effortlessly and without the need for external guidance.

However, many natural user interfaces (NUIs), while leveraging 'natural' interactions, often feel intuitive only after users have learned how to engage with them. For example, gesture-based systems may seem natural, but they often require initial learning or practice, where documentation or onboarding can be helpful.

5. Discussion and conclusions

The increasing prevalence of implicit interactions calls for the development of robust design criteria to ensure usability and effectiveness.

In this paper, we defined implicit interactions by distinguishing between the sensed task, which is sensed by the system, and the supported task, which the system supports based on the collected data. This distinction is essential for understanding how users engage with proactive systems and how implicit interactions unfold in real-world contexts.

Our work underscores the importance of carefully selecting the sensed task and accurately designing the system's sensing mechanism. This significantly contributes to maintaining key HCI usability principles such as matching the system with the real world, ensuring consistency, preventing errors, and supporting interaction personalization. Given that the sensed task is something the user would typically perform regardless of the implicit interaction, it is crucial that the system does not unduly influence or alter this task. We argue that if the design forces the user to modify their sensed task, it signals a deeper design issue.

Equally important is the selection and support of secondary tasks, which must adhere to usability principles such as error prevention, fostering a simple mental model, content personalization, consistency, and feedback. Poor support for secondary tasks, or failure to improve their effectiveness, suggests a flaw in the implicit interaction design.

The relationship between sensed and supported tasks must also be carefully considered in light of usability principles like system status visibility, user control and freedom and predictability. We proposed examples of situations where users might not want supported tasks to occur, or, conversely, cases where users may engage in the sensed task specifically to activate support for secondary tasks.

Ultimately, we argue that any failure to support secondary tasks effectively diminishes the overall value of the implicit interaction. If the secondary task's support fails, or even worsens the interaction, it highlights a fundamental flaw in the system's design.

In conclusion, as implicit interactions continue to evolve and become more prevalent, it is essential to address these design considerations to ensure that proactive systems align with user expectations and provide natural, effective experiences. Designers should reflect on these challenges when creating systems that seamlessly blend into users' lives, offering support without unnecessary disruption.

References

[1] D. Tennenhouse, Proactive computing, Commun. ACM 43 (2000) 43–50. URL: https://doi.org/10.1145/332833.332837. doi:10.1145/332833.332837.

[2] C. Meurisch, C. A. Mihale-Wilson, A. Hawlitschek, F. Giger, F. Müller, O. Hinz, M. Mühlhäuser, Exploring user expectations of proactive ai systems, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (2020). URL: https://doi.org/10.1145/3432193. doi:10.1145/3432193.

[3] Y. Deng, W. Lei, W. Lam, T.-S. Chua, A survey on proactive dialogue systems: problems, methods, and prospects, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23, 2023. URL: https://doi.org/10.24963/ijcai.2023/738. doi:10.24963/ijcai.2023/738.

[4] J. Nielsen, Heuristic evaluation, in: J. Nielsen, R. Mack (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, 1994.

[5] N. van Berkel, M. B. Skov, J. Kjeldskov, Human-ai interaction: intermittent, continuous, and proactive, Interactions 28 (2021) 67–71. URL: https://doi.org/10.1145/3486941. doi:10.1145/3486941.

[6] J. Grosinger, On proactive human-ai systems, in: International Workshop on Artificial Intelligence and Cognition, 2022. URL: https://api.semanticscholar.org/CorpusID:258845464.

[7] A. Schmidt, Implicit human computer interaction through context, Personal Technologies 4 (1999) 191–199. doi:10.1007/BF01324126.

[8] A. Schmidt, T. Herrmann, Intervention user interfaces: a new interaction paradigm for automated systems, Interactions 24 (2017) 40–45.

[9] G. Cockton, C. Gram, Design principles for interactive software, Chapman and Hall (Springer Science & Business Media), 1996.

[10] A. Dix, Beyond intention - pushing boundaries with incidental interaction, 2002. Conference on Building Bridges: Interdisciplinary Context-Sensitive Computing ; Conference date: 01-01-1900.
[11] B. Serim, G. Jacucci, Explicating "implicit interaction": An examination of the concept and challenges for research, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1–16. URL: https://doi.org/10.1145/3290605.3300647. doi:10.1145/3290605.3300647.
[12] Y. Deng, L. Liao, Z. Zheng, G. H. Yang, T.-S. Chua, Towards human-centered proactive conversational agents, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 807–818. URL: https://doi.org/10.1145/3626772.3657843. doi:10.1145/3626772.3657843.
[13] E. Panizzi, The seismocloud app: Your smartphone as a seismometer, in: Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 336–337. URL: https://doi.org/10.1145/2909132.2926070. doi:10.1145/2909132.2926070.
[14] E. Bassetti, A. Luciani, E. Panizzi, Ml classification of car parking with implicit interaction on the driver's smartphone, in: Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part III 18, Springer, 2021, pp. 291–299.
[15] A. Bisante, E. Panizzi, S. Zeppieri, Implicit interaction approach for car-related tasks on smartphone applications, in: Proceedings of the 2022 International Conference on Advanced Visual Interfaces, 2022, pp. 1–5.
[16] F. Nothdurft, S. Ultes, W. Minker, Finding appropriate interaction strategies for proactive dialogue systems—an open quest, 2015. URL: https://api.semanticscholar.org/CorpusID:41174018.
[17] F. Mueller, N. Semertzidis, J. Andres, J. Marshall, S. Benford, X. Li, L. Matjeka, Y. Mehta, Toward understanding the design of intertwined human–computer integrations, ACM Transactions on Computer-Human Interaction 30 (2023) 1–45. doi:10.1145/3590766.
[18] D. Bennett, O. Metatla, A. Roudaut, E. Mekler, How does hci understand human agency and autonomy?, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, 2023, pp. 1–18. doi:10.1145/3544548.3580651.
[19] A. Bisante, A. Dix, E. Panizzi, S. Zeppieri, To err is ai, in: Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter, CHItaly '23, Association for Computing Machinery, New York, NY, USA, 2023. URL: https://doi.org/10.1145/3605390.3605414. doi:10.1145/3605390.3605414.
[20] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al., Guidelines for human-ai interaction, in: Proceedings of the 2019 chi conference on human factors in computing systems, 2019, pp. 1–13.
[21] K. Helms, B. Brown, Design methods to investigate user experiences of implicit interactions, in: Proceedings of the 2016 ACM International Conference on Designing Interactive Systems, 2016, pp. 829–840.
[22] A. Dix, Activity modelling for low-intention interaction, in: The Handbook of Formal Methods in Human-Computer Interaction, Springer, 2017, pp. 183–210.
[23] D. Avola, P. Bottoni, M. Laureti, S. Levialdi, E. Panizzi, Managing groups and group annotations in madcow, in: Proceedings of the 6th International Conference on Databases in Networked Information Systems, DNIS'10, Springer-Verlag, Berlin, Heidelberg, 2010, p. 194–209. URL: https://doi.org/10.1007/978-3-642-12038-1_13. doi:10.1007/978-3-642-12038-1_13.
[24] J. Reason, Human error, Cambridge university press, 1990.
[25] D. A. Norman, The psychology of everyday things., Basic books, 1988. Revised version: The design of everyday thing, 2013.