Current Science
CS 3 (1), 266– 273 (2023)
Received 20 Oct 2022 | **Revised** 15 Nov 2022 | **Accepted** 15 Dec 2022 | **Online Available** 15 Jan 2023

Current Science

Current Science

OPEN ACCESS

ISSN (O) 2795-8639

ORIGINAL ARTICLE

Exploring Challenges of Developing Human-Centric Ai Systems for Ux Designers

Ayse Kok Arslan

Researcher Silicon, Valley Oxford Northern California Alumni San Francisco, Northern California, United States

Corresponding Author: Ayse Kok Arslan

Abstract: This study aims to improve the understanding of the challenges when it comes to creating human-centered AI systems for UX designers. After a discussion of current challenges faced by UX- and AI-related roles the study describes the limitations of existing practices which can become an obstacle for cross-disciplinary collaboration. Based on an analysis of existing HCI guidelines, the study discusses alternative standard software development workflows to bridge knowledge boundaries and highlight solutions for collaboration and co-design of HAI systems. The study also makes recommendations on how to implement HAI guidelines and develop required HAI training.

1. Introduction

UX roles involve interdisciplinary fields such as human psychology and design given the need for close collaboration with end-users to define system requirements. Software engineers skilled in computer programming implement those requirements within a system by means of highly controlled and abstracted knowledge to be translated into technical requirements for the software code [77].

This work aims to enhance both the understanding and challenges in creating human-centric AI systems. It suggests to improve the team-based approach for HAI (human-centered AI) based on an in-depth research of the following question:

What challenges do HAI designers and engineers face in creating HAI systems following the standard SoC process?

By highlighting the challenges faced by UX and AI roles in terms of developing AI systems in alignment with human values, the study explores the limitations of existing practices which could hinder cross-disciplinary collaboration despite being optimized for efficiency. Furthermore, it discusses alternatives to standard software development workflows and highlight solutions for collaboration and co-design of HAI systems.

Finally, this study aims to offer advice for technology companies to realize HAI guidelines and make recommendations for related training to increase an awareness of HAI.

Related Work

Human-Centered AI (HAI) frames AI as technology that, “augments the abilities of, addresses the societal needs of, and draws

inspiration from human beings” [67]. Based on this vision, research in HCI (human-computer interaction) and AI communities has characterized the following viewpoints [6, 18, 30, 90] to operationalize HAI:

(1) HAI systems offer greater capacity for human-like intelligent behavior,

(2) they dynamically learn and adapt their behavior over time through feedback and learning, and

(3) their outputs can be non-deterministic, making it difficult to align output presentation with end-user expectations [102].

Therefore, design knowledge for human-AI

systems is comprised of;

(1) understanding task characteristics, including type of goals and data representations,

(2) machine learning paradigms,

(3) human-AI interactions such as machine teaching, and

(4) AI-human interactions such as interpretability [28].

In order for the resulting human-AI system design to succeed, designers and engineers should closely collaborate. Typically, AI engineers and designers are situated in separate reporting structures (e.g: different physical environments and incentives), and professional exchanges occur less often between professional groups. Collaboration requires mutual respect for the expertise unique to each group.

In HCSD ("human-centered software development"), designers take a "UX first" approach to design the system's 'user interface' [87] which can be considered as the highest level module. While designers map end-user needs into interface design specifications engineers translate interface representation into implementation [84] to design the interface for AI-driven applications.

In taking a human-centric approach, UX designers need to provide inputs about AI model behavior by considering how the AI experience will integrate with end-user task workflow: what to automate, when to offer assistance, and when to maintain human control of tasks. Designers must also consider uncertainties in AI model outputs and design interface adaptations for explainability, failures, feedback, and hand-off [10, 93].

The ultimate aim in any AI model is to make "good" predictions. While no agent can be expected to excel in all imaginable control tasks, especially those far outside of its training distribution, training an agent which is generally capable on a large number of tasks is possible.

Researchers at Google AI (2022) hypothesize that ideally, an model should entail scaling data and computing parameters, continually broadening the training distribution while maintaining performance in order to unlock combinatorial generalization to new behaviors.

Rather than simply memorizing the data being fed into the model, the model should also be able to make good predictions on future samples, samples not seen before. Model evaluation is certainly not just the end point of any AI design pipeline and should also take the following into account:

- Estimating the predictive performance of a model on future (unseen) data
- Increasing the predictive performance by adjusting the learning algorithm and selecting the best performing model from a given hypothesis space
- Identifying the machine learning algorithm that is best-suited for the problem at hand; thus, comparing different algorithms to identify the best performing model within the algorithm's hypothesis space

Current practices in which the AI components are developed before envisioning the human user experience (i.e., AI-first design process) have led to AI systems that do not align with human needs (e.g., incorrect labeling [62], faulty facial recognition features [18], etc.). To overcome these challenges, several aspects of HAI design need to be incorporated throughout AI workflow, including identifying model requirements, data collection and labeling, features engineering, and model training [5, 42, 80].

Amershi et al. proposed a nine-stage software engineering workflow for machine learning that begins with specifying model requirements and subsequently, data collection, features engineering, and model development [5].

Subramonyam et al. proposed a HAI design process model for early-stage co-design [93]. This process may aid in catching emerging needs for hidden information in a just-in-time format rather than at its end. Designers can also create and collect leaky abstractions to help engineers (and users) envision how a final application may feel to use, and how design choices may differently impact different users.

Yang et al. study UX practitioners and their design processes for HAI. Through this investigation, they highlight challenges for realizing the double diamond UX process model for AI interface design [26, 99, 103]. As Yang's description of "designerly abstractions and exemplars [101]" suggests, leaky abstractions may be needed to address a wide variety of cases where lower-level detail is needed to inform and support interface and interaction components.

Leaky abstractions provide glimpses of the "other side," windows large enough to help but not overwhelm or take over their own perspectives. Leaky abstractions allow engineers to (1) communicate about needed training data characteristics for user interface design, (2) communicate model behavior for user experience design, and (3) evaluate the AI with end-users.

As AI technology is based in a data-intensive approach that challenges conventional UX design practices, sharing leaky abstractions allows designers and engineers to overcome knowledge blindness and engage in collaborative HAI design.

Project managers with a more holistic perspective may help facilitate cross-talk between engineers and designers. Yet, they should also take into account that not all UX designers are trained in these aspects of HAI systems due to the following reasons:

- First, UX designers lack the expertise to generate design ideas for incorporating AI in human tasks [32, 102].
- Given that AI takes a long time to build [101], rapid prototyping with ML through a "fail fast, fail often" approach characteristic of UX design is challenging for HAI [100].
- UX processes favor creativity and imagination of desired futures, which contradicts AI's emphasis on specificity and accuracy [98]. This introduces friction into the design thinking process for HAI systems.

Off-the-shelf ML systems and data generation tools can support designers as they investigate alternative designs for data-intensive AI. New tools are being created; for example, Proto-AI is a prototyping tool for designers that can directly invoke AI models and services, incorporate model outputs into interface designs, and iteratively and rapidly evaluate their design choices across diverse end-users and data.

In summary, previous studies tended to focus solely on data scientists [72, 73, 104] or designers [100, 101]. It remains an open question on how to handle abstraction in multidisciplinary collaboration between technical and non-technical roles. This work aims to address this gap.

Methods

To address research questions on collaborative HAI practices, various companies have offered recommendations for human-AI application design based on their internal design and development practices. Some examples include Microsoft's "Guidelines for Human-AI Interactions" [4, 7], Google's "People + AI Guidebook" [39], and Apple's "Human Interface Guidelines" [51].

As shown in Figure 1, the model consists of four main components, including (1) human mental models, (2) user interface, (3) AI models, and (4) training data. As indicated by the arrows, humans (and their mental model) are tightly linked to all other components to realize human-centered design.

Human mental-models target (1) understanding how end-users would perform a task on their own and the challenges they might face; that is, the task model; (2) understanding people's expectations about what the AI should do, and setting expectations for people about AI behavior, which we call the expectation model, and (3) identifying the best type of AI interaction experience given the situational context; namely, the interaction model.

User Interface guidelines target the software and hardware interface between end-users and AI. The recommendations center on lowering the gulf between execution and evaluation [74] by designing for (1) end-user inputs and AI outputs, (2) explainability, (3) feedback, and (4) failures and hand-offs.

Training data includes (1) planning data sources, (2) data collection, (3) labeling data, and (4) privacy,

security, and ethics of data. For instance, when planning data needs, guidelines recommend aligning them with the task model by asking what information an individual will use to perform the task on their own [39].

For data collection, the guidelines include (1) responsibly sourcing the data, (2) planning data collection to be representative of expected end-users, use cases, and context of use, (3) formatting data in ways that make sense to human users, and (4) collecting only the most essential information from end-users.

For labeling data, these guidelines focus on using proper labels; i.e., data labels must reflect the people's diversity and cultural context.

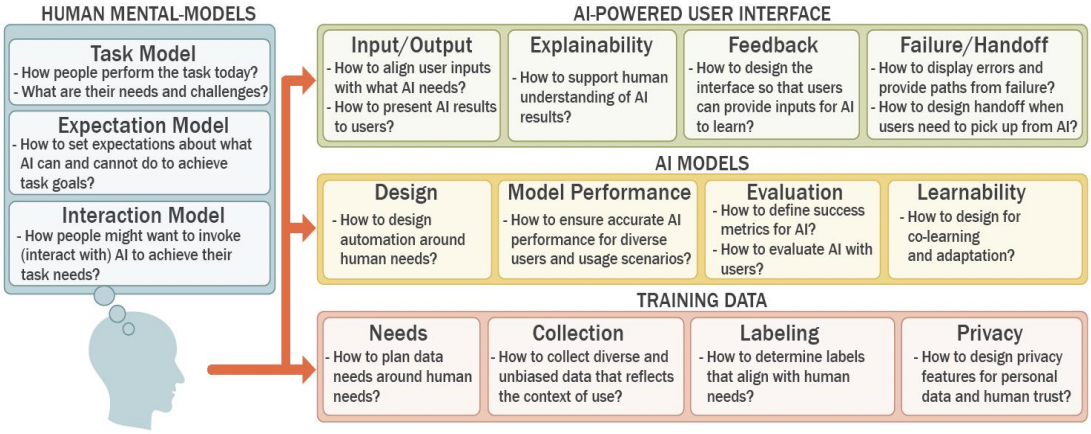

Figure1. Overview of HAI guidelines

Implementing these guidelines again requires that designers understand the AI's data needs and the types of computation that AI engineers will apply to the data.

When prototyping ML models, engineers create envisioning prototypes to demonstrate capabilities and potential uses to designers. In other cases, they work with design teams to 'align' model logic with interface designs by directly annotating over UI wireframes.

Conventional software design may be best accomplished with a "divide and conquer" process; but in cases where integrative expertise is needed, collaboration may require sharing key lower-level details (and not all). Because HAI development now requires fusing human needs within technical designs, points of intense collaboration across expertise roles will occur.

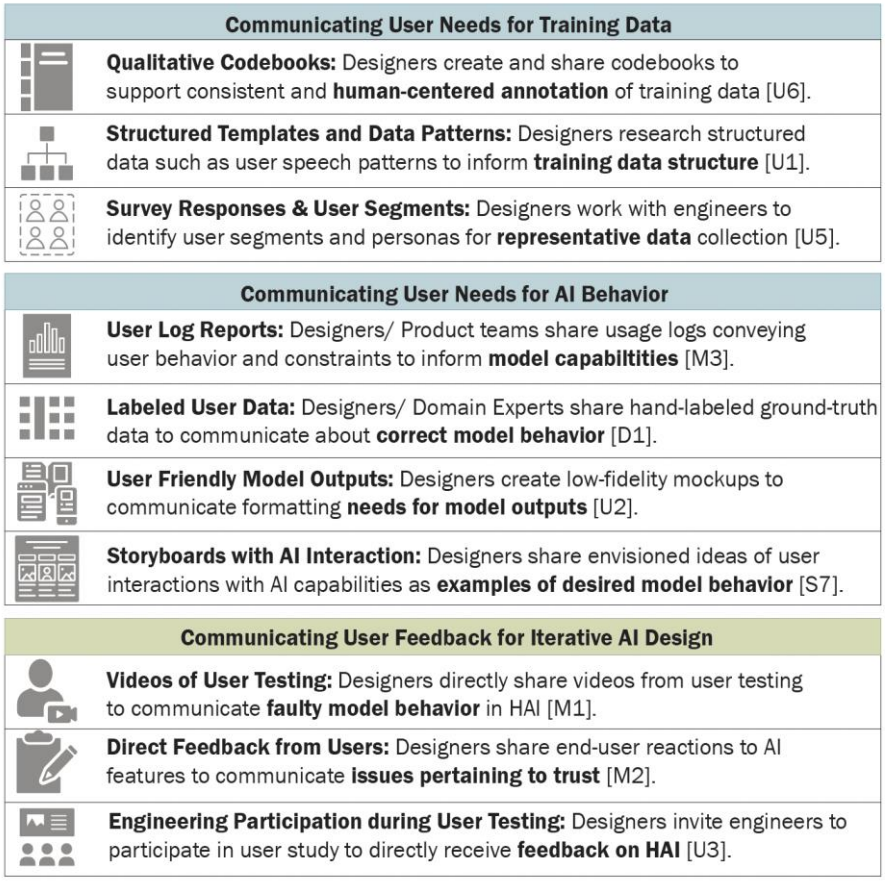

Figure2. Overview of low-level UX knowledge

As shown in Figure 2, successful teams delay system specifications through iterative prototyping and constant evaluation. In the early design stages, designers and engineers produce fuzzy design specifications with some aspects more concretely defined. By sharing those initial design artifacts including low level details, teams overcome knowledge blindness to align AI and UX, and then collaboratively assess, negotiate, and revise their design choices.

For instance, by sharing emerging AI behavior specifications, designers can evaluate assumptions and fit for end-users, update their own design representations for task workflows and interactions, and provide feedback for human-centered design of AI.

Such a workflow could address critical concerns around responsible AI design. Selbst and colleagues argue that social context information may be critical for some design considerations, so standard abstraction methods may require alteration for HAI design. To design AI systems with fairness in mind, teams need to collaboratively define fair performance by considering diverse stakeholders, contexts of use, and assessment criteria (i.e., disaggregated evaluation [14]).

On the other hand, an ethical framework for responsible AI design should also be taken into account by UX designers. Various suggestions have been made for how to address the increasingly global environments of contemporary engineering, although there has been little agreement about what global engineering ethics should be.

Within these debates, Zhu and Jesiek have identified and described four main approaches in terms of how global AI ethics has been conceived and taught:

(1) global ethical codes – outlining principles for ethical AI that would hold across cultures and nationalities;

(2) functionalist theories – where AI is conceived as a profession, such that it "functions" as a culture with a common ethical framework;

(3) cultural studies – exploring AI and technology vary by culture, and the ethical implications of these differences; and

(4) global ethics and justice – similar to global ethical codes, but concerned with ethical principles, in general, rather than ones specific to AI (Zhu & Jesiek, 2017).

While approaches 1, 2, and 4 are universalist in their aspirations, focused on formulating codes of ethics, theories, and curricula that would apply across cultures and nations, approach 3 could be described as particularist, tailoring their form and contents to different national and cultural traditions.

Luegenbiehl and Clancy have proposed a synthesis of the particularist and universalist approaches derived from the values of AI and evolved nature of human cognition, to be applied to and further refined in relation to case studies representing different technologies and cultural concerns, as a "bottom-up" approach (Clancy, 2021; Luegenbiehl & Clancy, 2017).

By sharing machine learning performance metrics and results (e.g., word error rate) with designers, software engineers can better align model-level performance with diverse user needs and use contexts. Despite increasing awareness, organizational goals and resource constraints continue to pose challenges for building responsible AI guidelines into current design practices [49, 64].

Recommendations

In addition, regular discussions about data—so central to defining AI capabilities—may be helpful to both groups even without specific review goals. "Data dives" might share current observations, consider what data might better inform choices, and review what is known about what users want. Providing space (in schedules and location) to inhabit the co-design process and build team familiarity will likely increase communication.

To support co-design practices, as indicated by Yang et al. [101] documentation, development, and regularization of formats for leaky abstractions may be helpful. Organizations might build these formats based on current team experiences where shared leaky abstractions have proven beneficial.

HCI should equip future UX practitioners with data-driven design tools and methods to facilitate

co-design. New toolkits and instructions can make HAI design accessible for students from differing backgrounds through supportive pedagogy and tools. Similarly, AI engineers should be trained to understand the importance of UX in AI development, create and share representations of AI behavior before implementation, and to co-design AI working across boundaries.

Moreover, an HAI curriculum should bring together individuals from varying backgrounds to engage in learning about team co-design throughout the engineering pipeline. Harris and colleagues list eight outcomes of AI ethics education: (1) stimulate ethical imagination; (2) identify ethical issues; (3) analyze and apply concepts; (4) take responsibilities seriously; (5) develop ethical sensitivity; (6) learn about technical, professional standards; (7) improve ethical, technical judgments; and (8) increase ethical willpower (Harris et al., 1996, pp. 93–94).

Based on these insights, the following methods can be implemented to fulfill these objectives:

- First, learning outcomes could be "meta-integrated." Meta-integration consists in creating activities that fulfill multiple learning outcomes so that: it would (1) not generate additional burdens on already packed AI curricula; and (2) provide more realistic and "spontaneous" learning environments for AI students, much closer to their actual future working environments.
- Second, AI curricula could be modelled on design curricula. Design curricula typically identify and track the effects of technology on different stakeholders throughout the design process (Norman, 2013; Zoltowski & Oakes, 2014). As a result, ethics would be included at the beginning and throughout the AI curricula (Civjan & Tooker, 2020; Spiekermann & Winkler, 2020; Van Grunsven et al., 2021).

These pedagogies and assessment tools can be strengthened through their broader use, for instance, by exploring the validity and reliability of assessment tools in cross-cultural contexts. Integrating global dimensions into AI ethics would not simply teach students practical skills but also allow students to broaden their scope, develop awareness of interconnectedness, and cultivate moral sympathy and creativity.

Limitations

An essential aspect of HAI design is to meet responsible AI criteria. While this study addresses the knowledge sharing and co-design practices necessary for various responsible AI criteria, future research should examine socio-technical practices and develop end-to-end strategies for enabling responsible AI. For instance, it can focus on how co-design will require 'leaks' to bridge fair performance metrics across AI and UX.

Lastly, this work studies the boundaries between UX designers and AI practitioners. Future work should investigate how new emerging roles in HAI (such as ML-Operations practitioners), role-specific guidelines, and improved incentive structures, can provide organizational support for responsible AI design.

Conclusion

In conventional software design, a clean separation of concerns between UX design and software implementation provides effective coordination and hand-off between designers and engineers on a team. This study offers an approach to collaboration that involves deferred specifications through iterative design and constant evaluation. It also made recommendations for practice and pedagogy to support the collaborative creation of human-AI applications.

As boundaries between designers and engineers introduce knowledge blindness about end-users and technology, this study highlights the possibilities and limits of AI to equip all stakeholders with an understanding on design uncertainties. By doing so, it also aims to help designers get aligned with engineers on AI models in the presence of uncertainty.

References

1. Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In International Conference on Machine Learning. PMLR, 60–69.
2. Philip Agre and Philip E Agre. 1997. Computation and human experience. Cambridge University Press.
3. Deloitte AI. 2019. Deloitte Insights. https://www2.deloitte.com/us/en/insights/deloitte-insights-magazine.html
4. Saleema Amershi. 2019. Guidelines for human-AI interaction design. https://www.

microsoft.com/en-us/research/blog/guidelines-for-human-aiinteraction-design/
5. Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: a case study. In Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice. IEEE Press, 291–300.
6. Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. Ai Magazine 35, 4 (2014), 105–120.
7. Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, 3.
8. Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. 2018. Software engineering challenges of deep learning. In 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 50–59.
9. Rob Ashmore, Radu Calinescu, and Colin Paterson. 2021. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. ACM Computing Surveys (CSUR) 54, 5 (2021), 1–39.
10. Jan Auernhammer. 2020. Human-centered AI: The role of Human-centered Design Research in the development of AI. (2020).
11. Julie Baca, Daniel Carruth, Elijah Davis, and Daniel Waddell. 2018. Merging the Cultures of Design and Engineering: A Case Study. In International Conference of Design, User Experience, and Usability. Springer, 628–641.
12. Balsamiq. 2021. balsamiq. https://balsamiq.com/
13. Thierry Bardini. 2000. Bootstrapping: Douglas Engelbart, coevolution, and the origins of personal computing. Stanford University Press.
14. Solon Barocas, Anhong Guo, Ece Kamar, Jacquelyn Krones, Meredith Ringel Morris, Jennifer Wortman Vaughan, Duncan Wadsworth, and Hanna Wallach. 2021. Designing Disaggregated Evaluations of AI Systems: Choices, Considerations, and Tradeoffs. arXiv preprint arXiv:2103.06076 (2021).
15. Eric PS Baumer. 2017. Toward human-centered algorithm design. Big Data & Society 4, 2 (2017), 2053951717718854.
16. Michel Beaudouin-Lafon and Wendy E Mackay. 2009. Prototyping tools and techniques. In Human-Computer Interaction. CRC Press, 137–160.
17. Melanie Birks, Ysanne Chapman, and Karen Francis. 2008. Memoing in qualitative research: Probing data and processes. Journal of research in nursing 13, 1 (2008), 68–75.
18. Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Conference on fairness, accountability and transparency. PMLR, 77–91.
19. Carrie J Cai and Philip J Guo. 2019. Software developers learning machine learning: Motivations, hurdles, and desires. In Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), VL/HCC, Vol. 19.
20. Paul R Carlile. 2002. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. Organization science 13, 4 (2002), 442–455.
21. Marcelo Cataldo and James D Herbsleb. 2010. Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. ACM, 65–72.
22. M Ceconello, D Spallazzo, and M Sciannamè. 2019. Design and AI: prospects for dialogue. (2019).
23. Herbert H Clark and Susan E Brennan. 1991. Grounding in communication. (1991).
24. Melvin E Conway. 1968. How do committees invent. Datamation 14, 4 (1968), 28–31.
25. Eric Corbett, Nathaniel Saul, and Meg Pirrung. [n.d.]. Interactive Machine Learning Heuristics. ([n. d.]).
26. Design Council. 2005. The ‘double diamond’design process model. Design Council (2005).

27. Cleidson RB de Souza, David Redmiles, Li-Te Cheng, David Millen, and John Patterson. 2004. Sometimes you need to see through walls: a field study of application programming interfaces. In Proceedings of the 2004 ACM conference on Computer supported cooperative work. ACM, 63–71.
28. Dominik Dellermann, Adrian Calma, Nikolaus Lipusch, ThorstenWeber, SaschaWeigel, and Philipp Ebel. 2019. The future of human-ai collaboration:a taxonomy of design knowledge for hybrid intelligence systems. (2019).
29. Norman K Denzin and Yvonna S Lincoln. 2011. The Sage handbook of qualitative research. sage.
30. Virginia Dignum. 2017. Responsible artificial intelligence: designing AI for human values. (2017).
31. Edsger W Dijkstra. 1972. The humble programmer. Commun. ACM 15, 10 (1972), 859–866.
32. Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. Ux design innovation: Challenges for working with machine learning as a design material. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM, 278–288.
33. Maria R Ebling. 1998. Translucent cache management for mobile computing. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.
34. Boris Ewenstein and Jennifer Whyte. 2009. Knowledge practices in design: the role of visual representations asepistemic objects'. Organization studies 30, 1 (2009), 07–30.
35. Gerhard Fischer. 2000. Symmetry of ignorance, social creativity, and meta-design. Knowledge-Based Systems 13, 7-8 (2000), 527–537.