Computational Assistance for User Interface Design: Smarter Generation and Evaluation of Design Ideas

Lena Hegemann
lena.hegemann@aalto.fi
Aalto University
Finland

Yue Jiang
yue.jiang@aalto.fi
Aalto University
Finland

Joon-Gi Shin
joongi.shin@aalto.fi
Aalto University
Finland

Yi-Chi Liao
yi-chi.liao@aalto.fi
Aalto University
Finland

Markku Laine
markku.laine@aalto.fi
Aalto University
Finland

Antti Oulasvirta
antti.oulasvirta@aalto.fi
Aalto University
Finland

ABSTRACT

This paper describes a lab demo by the User Interfaces group at Aalto University. The demo allows attendees to interactively experience recent research prototypes aiming to facilitate designers' creative and problem-solving capabilities in user interface (UI) design. Empirical work on designers suggests that UI design is challenging, partially because of the presence of very large design spaces, multiple and ill-defined objectives, design fixation and biases, as well as multiple requirements that need to to kept in mind. At the exhibition, members of the lab provide live demonstrations of six computational features, with a special focus on plug-ins created for Figma, a popular UI design tool. The demos draw from the group's latest research published at HCI conferences. They demonstrate how to interactively exploit machine learning methods ranging from deep nets to Bayesian inference and NLP. We also present our design approach and provide a summary of findings from empirical evaluations with designers.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Computational design, design tools, design-support systems, AI-assisted design, interaction design, UI design

ACM Reference Format:

Lena Hegemann, Yue Jiang, Joon-Gi Shin, Yi-Chi Liao, Markku Laine, and Antti Oulasvirta. 2023. Computational Assistance for User Interface Design: Smarter Generation and Evaluation of Design Ideas. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3544549.3583960

CHI EA '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3583960

1 INTRODUCTION

Recent developments in computational methods and design tools have allowed imagining new ways of assisting the creative practice of user interface (UI) designers. Methods like deep nets, optimization, and Bayesian inference can optimize designs and learn from user interactions and existing datasets. However, it is an open problem how to best integrate such methods into design tools, moreover in such a way that designers are willing to adopt and use them. Over years of HCI research on this topic, several requirements have been identified: (1) the designer needs to be in control and understand the consequences of assistance; (2) computational methods err, and designers should be offered a way to avoid errors or recover from them efficiently; (3) the overheads (costs) of accessing an intelligent feature should be minimized and kept low in proportion to the benefits to the designer; (4) designers should be given enough time to learn their own way of using intelligent features; and (5) designers should be allowed to iteratively refine problems and solutions.

This lab demo presents results form recent work of the User Interfaces group at Aalto University. The six demos show intelligent features aimed to address these requirements. They support different parts of the user-centered design process (see Figure 1).

1.1 Approach

Computational design is an exciting frontier combining insights and methods from ML, HCI, and Design. The demos presented here show how computational methods can be exploited to support key tasks in UI design, such as exploration, evaluation, and review of ideas. The techniques have been designed with the purpose of assisting not replacing or automating design work. To this end, we have engaged not only in designing new interaction techniques, nor ML methods, but also in empirical studies of designers. We believe that the value that designers obtain from an assistive feature depends on the accuracy of its suggestions, their timeliness, as well as the overheads of accessing it. The demos show how we have designed these features into existing design tools, making them available quickly and on demand without disrupting or overriding the designer. Emerging empirical evidence suggests several benefits to this approach: improved quality of end-results, increased cognitive resources saved for creative ideation, improved sensemaking, and improved consensus with stakeholders. The best way to understand these emerging techniques is by trying them out.

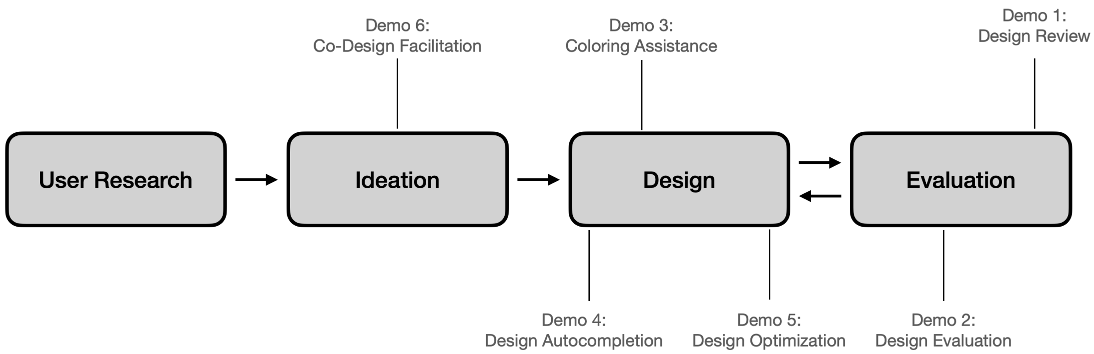

Figure 1: At the CHI'23 Interactivity Lab Demo, we present six computational features that assist designers in UI design in various parts of user-centered design.

2 EXHIBITION SETUP AND WALKTHROUGH

The lab demo consists of six demos, each presented on a small standing desk with a computer (the presenter's laptop). In addition, one of the demos uses a wrist-worn haptic display prototype, and another allows visitors to interact using their own phones. Each demo will have an associated poster explaining the main idea and providing links to more information. A larger monitor at the back of the exhibition area shows a video with a short introduction to all demos, inviting attendees to interact.

Attendees can enter the area from three different directions. There will be five presenters present. One will always be available to greet new arriving attendees and introduce them. Visitors can then choose to experience the demos as they wish. Attendees can try *Design Review* (Section 3) to analyze and fix different designs that have design guideline violations. With *Design Evaluation* (Section 4), they can inspect the designs to see how users are likely to perceive and experience them. Both Design Review and Design Evaluation are implemented as Figma plug-ins. *Coloring Assistance* (Section 5) shows another plug-in for exploring color related designs. At the *Design Autocompletion* (Section 6) table, attendees can edit GUI layouts and see how they can be autocompleted. At the *Design Optimization* (Section 7), the attendee goes through an optimization process and then experiences a few representative final designs using a wrist-worn haptic display. Finally, *Co-design Facilitation* (Section 8) introduces a chatbot-based co-design facilitator that attendees can interact with. They can use the Telegram platform on their own phones if they have the application installed. The booth will also provide summaries of empirical evaluations of these features.

3 DEMO 1: DESIGN REVIEW

Design Review is a design assistant that helps designers to improve the consistency, compliance, and quality of their designs. It uses rule-based AI and optimization under the hood. At the demo booth, attendees can sketch UI designs, or alternatively, use example designs and interact with the feature to identify and fix guideline violations. The assistant identifies design guideline violations and proposes fixes to them automatically. During the design process, the system (1) evaluates a design against selected design system guidelines, (2) reports and highlights detected guideline violations, and (3) assists the designer in fixing them (Figure 2-a). The final decision on whether to apply the proposed fixes is always up to the designer.

Design Review is designed to support realistic design systems. For example, Material Design 3[1], which we support, includes well over one hundred design guidelines. Compared to other design review systems, such as *UIS-Hunter* and *Lint for Sketch*[2], our assistant better integrates with today's popular UI design tools[3], offers a more scalable architecture, and enables automated fixing of violations with a click of a button. The benefits of the assistant are three-fold: It (1) facilitates the adoption of design systems in design teams, (2) makes designs more consistent and intuitive, and (3) reduces time spent in time-consuming design review meetings.

4 DEMO 2: DESIGN EVALUATION

Design Evaluation is a Figma plug-in that assists designers in estimating how users perceive designs. The system uses state-of-the-art predictive models [4] for visual saliency maps (heatmap) and scanpaths (gazeplot) that are applicable to a broad range of user interface types. Figure 2-b shows an example of heatmap and gazeplot predictions as displayed in our plug-in window. Attendees can try out the feature by editing example designs or creating new ones in Figma and seeing the predictions in near real-time. The assistant can also visualize predictions by other computational evaluation metrics, such as those we have published in the open-sourced Aalto Interface Metrics (AIM)[4] server [6].

5 DEMO 3: COLORING ASSISTANCE

CoColor is a novel interaction technique that assists color design [3]. Attendees can interact with the coloring assistant in three color related design spaces. The first is about choosing the focus of coloring. For instance, this could be a particular mood, action, or product image that the design should promote. The second is concerned with the color palette; that is, the set of colors used in the design. The third relates to colorizations; that is, the assignment

[1] Material Design 3, https://m3.material.io/
[2] Lint for Sketch, https://github.com/saranshsolanki/sketch-lint
[3] 2022 Design Tools Survey, https://uxtools.co/survey/2022/
[4] Aalto Interface Metrics (AIM), https://interfacemetrics.aalto.fi

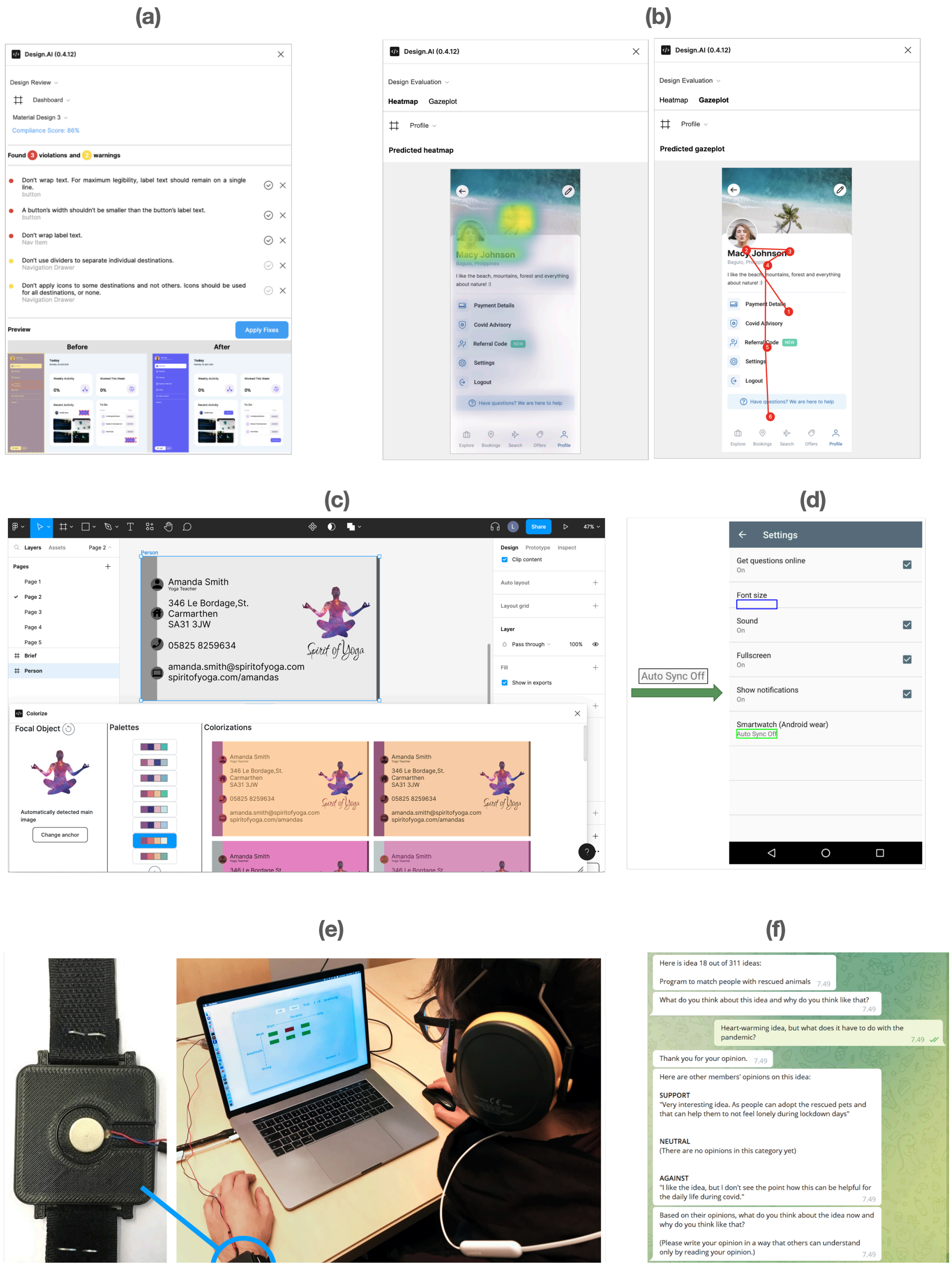

Figure 2: At the CHI'23 Interactivity Lab Demo, we present six computational features that assist designers in UI design: (a) the Design Review feature, integrated into Figma, detects violations of design guidelines (think: Material Design) and proposes how to fix them; (b) the Design Evaluation feature evaluates UI designs using metrics and predictive models, such as visual saliency; (c) the Coloring Assistance feature explores and proposes alternative colorings of a design; (d) the Design Autocompletion feature proposes how to place a given UI component on a given UI frame; (e) the Design Optimization feature proposes how to optimize given parameters in a design for designer-specified objectives; (f) the Co-design Chatbot allows a team of designers to ideate and refine design ideas together in hybrid settings.

of the colors to elements and rendering of the resulting colored design.

CoColor supports the rapid exploration and iteration through these three design spaces. It proposes automated transitions between them but yet allows edits by the designers at every step. The effects of the edits become visible in the design at hand immediately. See Figure 2-c for a screenshot of the Figma plug-in. Prior techniques either fragmented the exploration of the three design spaces or automated the full pipeline, making iteration challenging. A user study with 16 professional designers showed that CoColor makes color exploration easier and it was considered useful by most participants [3].

6 DEMO 4: DESIGN AUTOCOMPLETION

In this demo, attendees can interact with an autocompletion feature. It uses a novel graph representation approach that assists the scaling and placement of UI components on UI frames (see Figure 2-d). Autocompleting UI wireframes is hard because of the computational complexity of evaluating possible combinations of UI element positions. In this demo, given a UI frame, our method automatically generates in-place suggestions shown as overlays in a design tool. The designer can simply 'throw' elements to the frame, which then snap to desired locations, saving the designer from time-consuming pixel-level editing.

The demo leverages a deep net based approach that uses heterogeneous graph representations (here: GNNs) to represent UI layouts. It combines both UI element information and layout constraints. Constraint-based layout models have been widely used in graphical user interface layouts (our previous work: [5]). The novelty here is that we can train the GNN to respect a particular style of a dataset while taking into account both textual and graphical tendencies. The demo will be set up on a computer with a screen and mouse. The implementation in a UI editor allows attendees to edit frames and get real-time predictions for completions.

7 DEMO 5: DESIGN OPTIMIZATION

This demo presents a novel machine learning-guided workflow for efficiently exploring design spaces [1]. Exploring a vast design space while balancing several design objectives is a challenge for designers. Our workflow is suitable for well-defined design cases toward the end of a design project, where the best possible design must be found. Given design parameters, the algorithm guides the designer to consider design candidates. The attendees will experience the workflow and can interact with the multi-objective Bayesian optimization method. They can explore so-called Pareto-optimal designs by trying them out.

This demo takes the design of a single-tactor tactile display as an example. Optimizing information transmission via haptic displays has been an important and persistent goal for haptics researchers. However, this design task depends on not just one performance metric but multiple, e.g., information throughput and recognition accuracy. A strategy for conveying information via such a display is to encode messages by unique combinations of vibration duration and amplitude. The display can convey more messages by increasing the number of unique combinations, but this also makes recognizing tactile cues more difficult. In this demo, Bayesian optimization proposes a set of vibration cues in each iteration, and the attendee evaluates them (Figure 2-e). Based on the evaluation, the optimizer proposes the next set of vibration cues, attempting to identify the Pareto-optimal designs as efficiently as possible. We show the designs with the best trade-offs for attendees to explore.

8 DEMO 6: CO-DESIGN FACILITATION

Converging early-stage design ideas and forming a consensus on a few design directions is a challenge. In this demo, we show how chatbots can facilitate idea generation and selection in co-design projects (Figure 2-f). Facilitating co-design is normally done without computational support. It requires high effort from human facilitators to engage with individual stakeholders and guide them to design together. In this demo, attendees get to ideate collaboratively through our chatbots. They guide participants to build on each other's ideas (i) by showing other users' ideas (i.e., inspirations) and (ii) by suggesting what users can ideate from the inspirations (i.e., ideation methods). They select inspirations and ideation methods while adapting to individuals' behavior, hence users can generate and select more helpful ideas in the end.

The demo builds on and extends our earlier work in this domain [7]. In the demo, we show how to use Multi-Armed Bandit (MAB) and an NLP-based semantic similarity when designing the chatbot's dialogue. The chatbot is good at presenting similar and dissimilar ideas to participants. It can inspire individual users to ideate more, integrating collaborators' viewpoints into their ideas.

9 DISCUSSION AND CONCLUSIONS

Computational methods, especially modern AI methods like LLMs, are contentious. On the one hand, they are perceived as stealing intellectual property and threaten to automate designers' work. On the other hand, they can be perceived as an opportunity to boost creativity. Our lab's motto is "Computational Superpowers for Designers". We want to design methods that facilitate their creative and problem-solving capacity. Our studies have shown that even if an AI-driven method can improve the "objective" quality of design outcomes, this does not trivially translate to higher impact of design, if it threatens the felt agency and ownership of the designer [1]. This lab demo shows that HCI research can identify interaction techniques that are acceptable and useful and successfully integrate them to tools. It shows how computational approaches can boost mundane, "pixel-level" work while keeping the designer in charge. Designers focus more on those aspects of design that are out of reach of computational methods, like creative ideation, sensemaking, positioning work, and reflection. This is how, we believe, computational methods can help designers create more equitable, accessible, and sustainable designs. However, it remains to be seen if they in fact help designers when possible second-order effects are factored in.

Besides long-term studies of these systems, one outstanding question for HCI is *shared control* in design tools. An intelligent system should have 'just the right level' of agency. In other words, it should be available when needed but also able to intervene when needed. It should be able to work at what ever level of control is needed, from low-(pixels) to high-level aspects. However, such interactions must be done while being sensitive to the designer, the situation, and the design domain [2]. To that end, we are studying interaction techniques for mixed-initiative interaction with AI, where the system could intervene when needed (our earlier work: [8]). However, these problems can only be solved in a transdisciplinary effort; a point our demo will make.

REFERENCES

[1] Liwei Chan, Yi-Chi Liao, George B Mo, John J Dudley, Chun-Lien Cheng, Per Ola Kristensson, and Antti Oulasvirta. 2022. Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques. In *CHI Conference on Human Factors in Computing Systems*. 1–14.

[2] Sebastiaan De Peuter, Antti Oulasvirta, and Samuel Kaski. 2021. Toward AI Assistants That Let Designers Design. *arXiv preprint arXiv:2107.13074* (2021).

[3] Lena Hegemann, Niraj Ramesh Dayama, Abhishek Iyer, Erfan Farhadi, Ekaterina Marchenko, and Antti Oulasvirta. 2023. CoColor: Interactive Exploration of Color Designs. In *28th International Conference on Intelligent User Interfaces* (Sydney, Australia).

[4] Yue Jiang, Luis A. Leiva, Hamed Rezazadegan Tavakoli, Paul R. B. Houssel, Julia Kylmälä, and Antti Oulasvirta. 2023. UEyes: Understanding Visual Saliency across User Interface Types. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA.

[5] Yue Jiang, Wolfgang Stuerzlinger, Matthias Zwicker, and Christof Lutteroth. 2020. ORCSolver: An Efficient Solver for Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376610

[6] Antti Oulasvirta, Samuli De Pascale, Janin Koch, Thomas Langerak, Jussi Jokinen, Kashyap Todi, Markku Laine, Manoj Kristhombuge, Yuxi Zhu, Aliaksei Miniukovich, et al. 2018. Aalto Interface Metrics (AIM) A Service and Codebase for Computational GUI Evaluation. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. 16–19.

[7] Joongi Shin, Michael A Hedderich, AndréS Lucero, and Antti Oulasvirta. 2022. Chatbots Facilitating Consensus-Building in Asynchronous Co-Design. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[8] Kashyap Todi, Gilles Bailly, Luis Leiva, and Antti Oulasvirta. 2021. Adapting user interfaces with model-based reinforcement learning. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.