

---

This is an electronic reprint of the original article.  
This reprint may differ from the original in pagination and typographic detail.

Oulasvirta, Antti

## Can computers design interaction?

*Published in:*

EICS 2016 - 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems

*DOI:*

[10.1145/2933242.2948131](https://doi.org/10.1145/2933242.2948131)

Published: 21/06/2016

*Document Version*

Publisher's PDF, also known as Version of record

*Published under the following license:*

Other

*Please cite the original version:*

Oulasvirta, A. (2016). Can computers design interaction? In *EICS 2016 - 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 1-2). ACM. <https://doi.org/10.1145/2933242.2948131>

---

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# Can Computers Design Interaction?

Antti Oulasvirta  
Aalto University  
antti.oulasvirta@aalto.fi

## ABSTRACT

Algorithms have revolutionized almost every field of manufacturing and engineering. Is interaction design the next? This talk gives an overview of what future holds for optimization methods in interaction design. I introduce the idea of using predictive models and simulations of end-user behavior in combinatorial optimization of user interfaces. I demonstrate it with an interactive layout optimizer and provide an overview of research results. I tell about the models we use, the limitations of this approach, how it fits the HCI engineering cycle, and how we validate and verify this approach. To conclude, I provoke a critical discussion on the potentials and pitfalls of this approach.

## Categories and Subject Descriptors

H.5.m. [Information Interfaces and Presentation (e.g. HCI)]: Miscellaneous

## Keywords

User interface design; combinatorial optimization; model-based design; behavioral modeling.

## 1. INTRODUCTION

In this talk I discuss a milestone that is within our reach in the design of user interfaces (UIs). Combinatorial optimization methods can search and propose a UI design that is not only appealing-looking but actually usable. UIs for mobile apps can be generated, such that users can find items quickly and perceive an UI as aesthetically pleasing. A programmer can ask an optimizer to design a full menu system, including its hierarchy, groups, and shortcuts, simply by listing its commands. The prospect of routinely generating solutions to some of the most widely used types of UIs, such as graphical UIs, menus, web pages, mobile apps, and visualizations now looks realistic. Scarcely 15 years ago, in the area of combinatorial optimization of UIs, the keyboard layout was the prime—and virtually only—case with empirically demonstrated benefits to end users.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*EICS'16 June 21-24, 2016, Brussels, Belgium*

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4322-0/16/06.

DOI: <http://dx.doi.org/10.1145/2933242.2948131>

Reaching this milestone would imply a paradigm change in the way we design UIs. Optimization and operations research, or “the science of better”, have revolutionized almost every field of engineering design and management, so why not UI design? In UI design too, optimization may not only improve the quality of solutions but also allow us to solve hard, previously unsolved problems. It would offer a flexible and powerful toolbox for designers’ problem-solving. They are not limited to finding a single “best” design; they explore the space for surprising alternatives too, a hallmark of computational creativity.

Critically, computational design follows from first principles. Unlike in our day-to-day user-centered design, those principles are explicated, scrutinizable, and can be questioned. But if you agree with the design task and its assumptions, you must agree also with the outcome. Moreover, unlike any other method conceived for UI design, optimization methods also offer a greater than zero chance of finding the optimal design. In fact, the class of exact optimization methods even offers mathematical guarantees for solutions. They can inform a designer by stating how far from the optimum the current design is. These benefits would be remarkably valuable for interaction designers.

In the talk, I compare this with the present user-centered design paradigm. It rests much more on human creativity, craft, insight, experience, empirical data, and persistence. It is through-and-through heuristic and iterative, and it emphasizes sense-making, empathy, and creation of meaning as objectives. While it is at its best in the ideation of UI concepts, optimization methods can complement it—particularly the problem-solving aspects of UI design.

## 2. RECENT ACHIEVEMENTS

In the talk, I review the history of computational approaches to UI design. The idea of searching a design space systematically for a better UI can be traced to August Dvorak, a psychologist famous for the Dvorak Simplified Keyboard (DSK). Unlike most psychologists, he was not content with pooling empirical results; he applied his findings to redesign the keyboard manually. The idea of using computational methods to this end, however, was discovered much later—and more or less independently in computer science, cognitive science, and HCI.

Unfortunately, progress was limited by inability to predict how users may use and experience a design, and by inability to formulate design tasks algorithmically. In the rest of the talk, I define and review model-based user interface optimization as an application of engineering optimization for discrete design problems: A UI optimization task consists of a finite set of candidate design, an objective function, and constraints. I discuss three technical advances that are now making this possible. Firstly, we now can define not only keyboard layout problems but complex visuospatial design tasks.

Secondly, whereas previously the use of predictive models of user behavior in algorithmic design was limited to simple motor laws, we now can address core issues in UI design, such as perception, visual attention, learning, and even individual differences. Third, we know much better how designers can express knowledge to a computer in a way that is compatible with how they think and work.

I argue that the most significant obstacle to progress has been not algorithmic but modeling-related. Extending from simple motor performance models to perceptual and cognitive factors has finally pushed the envelope in UI optimization. We can now optimize some of the most commonly used types of user interfaces.

### 3. FUTURE OF MODEL-BASED UI OPTIMIZATION

I conclude the talk with an overview of potentials and challenges.

On the one hand, the implications for the profession and business of user interface design can only be speculated. Optimization enables complete novices to obtain good designs without a professional designer in the loop. Designers stand to benefit, too, especially from the ability to improve low-level UI design. But it may not only improve quality but free resources for focusing more on the “un-computable” or “wicked” aspects of design—creative problems and conundra involving incomplete or contradictory knowledge, a large number of stakeholders who differ in their opinions, and severe constraints. More importantly, for the first time in the history of UI design, one can talk about it as an engineering discipline, discussing the robustness of a design or its sensitivity to assumptions. What if a designer could show proof that her design is 5% better than a competitor’s? What if the designer could show that there is *no* feasible solution to a problem? For the first time, computational and engineering sciences will have a fully legitimate role in the design of UIs, not just their implementation. This will create a new frontier in HCI where the focus is on representing design problems and design knowledge in code and efficiently solving the computation problems thus expressed.

On the other hand, however, we are still a far cry from this vision. Many challenges arise. How far can we push the envelope in modeling human behavior? While our work has shown success with basic sensory-motor models and is expanding to consider cognitive aspects, there is a long way to go to address social and cultural factors. How can we use big data in optimization? Modern design is driven by data analysis and online experiments. Perhaps such data could parametrize predictive models. The outcome would be a “self-designing UI” that would not adapt a feature blindly but use models to weigh the achievable benefits against the costs of changing the UI.

How well can we capture designers’ decisions formally? We have found uses that address classic topics such as assignment and packing, but are there yet unknown problem types? Perhaps the most startling proposition for design research is that the essence of design activity, which has been considered complex and nuanced, may, in fact, be expressed algorithmically.

What about modern optimization methods? Can they stretch to realistic UI design problems? We are talking about multiple design decisions, numerous objectives, missing or incomplete information, and very large search spaces. These issues exist alongside the need to produce good results at interactive rates in a design tool. While there are solutions to each of these problems individually, practical deployment requires finding ways to put them together.

Finally, what about creativity? While there is ongoing work on both human creativity and computational creativity, we have to find concepts that support the joint creativity of the designer–computer

system.

### 4. BIOGRAPHY

Antti Oulasvirta is a cognitive scientist researching human–computer interaction. He is an Associate Professor at Aalto University where he leads the User Interfaces group. Before joining Aalto, he was a group leader at the Max Planck Institute for Informatics and a postdoc at UC Berkeley. Dr. Oulasvirta has been awarded eight Best Paper Awards and Honorable Mentions at the CHI conference. He recently received an ERC Starting Grant to investigate model-based optimization of user interfaces.

### 5. ACKNOWLEDGMENTS

This work has received funding from the Academy of Finland project COMPUTED and the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement 637991).