



It's Time to Rediscover HCI Models

Insights

- Modeling has improved significantly since GOMS and Fitts's law.
- We have a new bedrock in powerful computational models that can explain and predict behavior with higher fidelity and address a broader scope beyond point-and-click interfaces.
- The most far-reaching development is that we have learned how to use models to drive the algorithmic generation and adaptation of UIs.

During the formative years of HCI, Stuart Card asserted that the tortoise of accumulative science would outrun the hare of intuitive design. It's time to look at the tortoise again.

It's commonly held, and corroborated by our textbooks, that modeling in HCI started and ceased with GOMS (goals, operators, methods, and selection) and Fitts's law. I argue that the tortoise has crept forward and that there are many reasons to get excited again [1]. We have a new bedrock in powerful computational models that can explain and predict behavior

with higher fidelity and address a broader scope beyond point-and-click UIs. But perhaps the most far-reaching development is that we have learned how to use models to drive the algorithmic generation and adaptation of UIs.

Modeling covers more ground and depth. The leading psychologists of the 1980s promoted *universal* theories of cognition, an idea that influenced the HCI community. Around the turn of the millennium, the mindset had changed: Modeling in HCI requires flexibility instead of unified architectures. We need to be able to scope a problem, choose

THE HCI MODELS SUPERMARKET

Learning equips an interface with the ability to adapt executable functionality to data, typically by optimizing parameters of a model to minimize a loss function. The flagship application in HCI is input recognition, but others extend to generative design.

- **Example:** So-called probabilistic grammars can learn design patterns in the wild and be used to generate new exemplars [4].

Probabilistic modeling has emerged as the go-to approach for interaction involving inference and reasoning based on noisy data. Language and touch models are used to decode the intended input of a user in intelligent text entry. Modern probabilistic inference methods allow very complex cognitive models to be fit to data.

- **Example:** Parameters of the vision system of a user can be inferred from menu-click logs by fitting a cognitive model with Bayesian optimization [5].

Bounded agents assume that user behavior is constrained by the structure of the environment and design, as well as the user's capacity and resource limitations.

- **Example:** The information-foraging model was used to explain (Zipfian) distributions in page hits observed on websites [6].

Computational rationality extends bounded agents to the case with uncertainty and delayed gratification, and estimates a user's behavioral strategy with methods like reinforcement learning.

- **Example:** The learning of element locations in a GUI can be explained by cognition that learns to use its perceptual and memory resources with experience [7].

State machines represent interaction events as discrete states and transitions. These permit rigorous analysis of the interface properties, such as reachability of functionality, critical paths, bottlenecks, and unnecessary actions.

- **Example:** Medical devices are assessed for safety (e.g., is feedback always provided for a critical action?) by checking UI code using logic [8].

Control theory models moment-by-moment interaction as a control problem: A controller must decide how to change its outputs to get to a goal. Applications in HCI so far have focused on input interaction, such as pointing, but there are multiple applications outside HCI.

- **Example:** A model of steering in driving [9].

Biomechanics models the mechanics of biological organisms. In HCI, it looks at interaction from a physical perspective and can help identify movements with the most effort or ergonomic risk.

- **Example:** The biomechanical model of human body predicts movement performance and muscle effort required by novel input methods [10].

the most suitable model class, and construct the model to our needs.

Unbeknownst to many, a rich repertoire of models has become available since GOMS. The first sidebar surveys the HCI Models Supermarket. The Supermarket covers parts of interaction that were poorly covered by the classics, including motor control, biomechanics, decision making, learning, planning, navigation, error handling—and even collaboration, aesthetics, and emotion. Much of this progress was not visible to the HCI community, occurring in

fields such as AI, control, machine learning, and cognitive science.

The two initial model families, mathematical and information processing-based, have stepped forward too. There are at least a hundred papers on aimed-movement models. Information foraging theory has successfully addressed central forms of decision making and navigation in complex information environments in terms of rational behavior. Information-processing models also evolved. The modern cognitive model is a hybrid modular architecture best exemplified

by ACT-R. It offers specialized symbolic and subsymbolic modules on top of a general architecture that has some built-in learning capabilities. This model has been applied to a number of HCI applications, including very complex ones like piloting. These either build on ACT-R or adopt modules from it for specialized models.

Also, a new category of cognitive models has emerged. *Computational rationality* approaches human behavior as *optimal adaptation to bounds under uncertainty*. The bounds are not only those of the task, but also those of the agent's own cognitive resources. Machine learning, such as reinforcement learning, can be used to find human-like behavioral policies for the agent. This elegant approach predicts the *emergence* of behaviors as a function of design and context, and, significantly, does not require prescribing a task solution like the classic information-processing models. Computational rationality is a candidate for explaining situated interaction and is starting to shed light on the natural intelligence humans show in interactive tasks.

Models analyze and generate designs. Stuart Card and colleagues imagined HCI would become like the field of operations research, where complex systemic problems are solved via modeling and optimization. Alas, the early models were not utilized beyond the evaluation of designs. This left a weak spot for critics.

There were serious attempts to make models more actionable. In CogTool, UI screenshots were uploaded and user tasks demonstrated, which were then evaluated with a cognitive model. The issue is that no hint is given on how to *improve* the design, only where time is spent, what taxes users' memory, or where they may err. Models can also be used to *analyze* qualities such as the robustness and scalability of an interface—for example what happens when functionality gets more numerous, canvas size changes, or users become more numerous or change their tasks. Models can also be integrated into design tools to assist designers with

When models cannot contact data from real world, they become hard to falsify and therefore hard to evolve.

analyses. Menu designs, for example, can be analyzed in real time to show suboptimal item placements or suitability for user groups [2].

But perhaps the most important development since the 1980s is the deployment of models as objective functions in computational design. The development of this idea can be traced along the footsteps of AI, from rule-based systems to logic, state machines, optimization, clustering, probabilistic models, and deep learning. Presently there are a number of interface types that can be generated, from keyboards to menus to visualizations and GUIs [1]. Logic and state machines have been used to verify interactive properties of safety-critical designs.

Models empower designers.

Since the 1990s, HCI has explored interactive tools that use models and algorithms to *assist* designers. They can augment and partner with designers via concepts like the Interactive Example Gallery: An algorithm detects the designer's task and presents design suggestions in a side panel. Designers are good at recognizing good solutions, though it may be much harder to produce one from scratch. The designer stays in control and can ignore the suggestions.

We have new capabilities in data-driven and gray-box modeling. While classic modeling in HCI started from first principles such as limits of cognition or the motor system, with machine learning there is now another breed. Models can be learned from labeled or unlabeled interaction data. One can think of this as a spectrum that goes from theory-driven *white-box models* to data-driven *black-box models*. Black-box models, for instance deep learning, can learn a distribution of a complex dataset to make predictions, classify, and generate. While they are less interpretable and controllable than white-box models, they tend to come with a much higher learning capability. *Gray-box models* are an interesting in-between creature. Some parameters are learned from data. Either the structure of the model is informed by theory (light gray) or a black-box model is trained with synthetic data generated with a white-box model

(dark gray). SUPPLE exemplifies a (very) light-gray model [3]. It started the field of ability-based design optimization, where GUI designs are algorithmically customized according to an individual's perceptual-motor characteristics.

We can fit complex models to user data. At the time of GOMS, modeling was approached mainly as a *forward* but not as an *inverse* problem. In forward modeling, we construct a model in order to generate human-like data. In inverse modeling, we go from data to model parameters. In other words, we find ("fit") the model parameters that most closely reproduce observed data. At the time of GOMS, regression models could be fit to

data (e.g., by using Least Squares), but more complex models were out of reach. This posed a barrier to progress that was not recognized at the time. When models cannot contact data from real world, they become hard to falsify and therefore hard to evolve.

Mathematically mature methods for model selection and parameter fitting have emerged since the 1990s and are becoming central in cognitive science and elsewhere. Black-box methods, gradient-based methods, and probabilistic methods are part of the modern toolbox. One powerful method is Bayesian optimization, which uses an inexpensive surrogate (proxy) model for approximating the model

OLD CRITIQUES, NEW RESPONSES

Some longstanding critiques of modeling can be revisited in light of recent advances:

Critique: "Models are limited to point-and-click UIs."

R: We have seen tremendous increase in scope since the 1990s covering, among others, VR and AR, the Web, mobile apps, conversational AI, social interaction, and novel forms of input techniques.

C: "Modeling is too arcane to be useful in design."

R: Perhaps the most exciting aspects of modeling are that they can run adaptive interfaces, generate new designs, and help designers. HCI research is pivotal in making the most out of them in design practice.

C: "Models are inherently unable to account for social, embodied, and situated aspects of behavior."

R: The HCI Model Supermarket now covers much more of the reality outside the desktop, including locomotion, multitasking, and aspects of social interaction. Moreover, we have new tools in learning and probabilistic reasoning that provide a powerful basis for working with very complex and uncertain phenomena. With those we are starting to understand the emergence of behavior as a function of individual-, context-, and design-related factors. Although we are nowhere near a satisfactory account, we are starting to see structure in apparent messiness.

C: "Models are hard to construct."

R: Modeling will always be hard because it can be scaled up. However, getting started is easy. There are a number of libraries and tools available, for example for control (SimuLink), optimization (scipy, Gurobi, CPLEX), learning (scikit-learn, PyTorch), probabilistic models (Stan, GPYOpt), and biomechanics (OpenSim, MuJoCo).

C: "Modeling is inherently flawed because the mind is not a computational mechanism."

R: Some of the most exciting advances in AI build on the computational account of the mind. We have not yet met hard limits to the approach. However, a modeler does not need to subscribe to the stance of mind as computation.

C: "Why bother? Usability testing and interaction design will get the job done."

R: If the problem is well-defined and easy, and neither thorough understanding nor a shared knowledge base is important, why not just wing it? But if this not the case, investing in modeling will offer long-term benefits via better science, improved quality of solutions and, with algorithmic design, higher cost efficiency.

C: "Modeling subscribes to a scientific stance, but HCI also needs to construct: to innovate and design."

R: Similar to other many other successful areas of applied science with constructive goals, such as in engineering and medicine, modeling can strongly support ideation, analysis, and the generation of solutions. It does not preclude other types of activities. HCI is in a unique position to identify efficient methods to exploit models to support human creativity and problem-solving.

fit across the parameter space. It is suitable for the inherently stochastic and expensive computations we have with HCI models. Moreover, we learn which parameter combinations are most plausible given the data by examining the so-called posterior distribution. What this means is that we have better tools to make models of particular users and contexts and thereby evolve them.

We know better what makes a good model. The criteria for what counts as a good model have developed. At the time of GOMS, and admittedly even today, researchers fixated on *descriptive adequacy*, often measured with *R-squared*. Obsession with minor increments to model fit may have fostered a false sense of progress. Model fits can be “hacked” as easily as in other sciences and do not communicate how general or useful the model is.

Outside HCI, many relevant criteria have surfaced. A *practical* model has minimum sufficient complexity for the problem. It generalizes beyond observation data to unseen circumstances, such as new user groups, contexts, and designs. Statistical methods are now available for such criteria, for example metrics based on information theory and the Bayes theorem, as well as cross-validation methods. Bayesian statistics are available as diagnostics tools for threats like overfitting and outliers. Admittedly, HCI has a long way to go to develop its own criteria, but much of the groundwork is available.

Modeling complements, not replaces, other methodologies. It is vital to rectify an unfortunate misunderstanding: Modeling is *not* an all-or-nothing choice, nor is it in inherent opposition with mainstream views of HCI. Interaction design is concerned with

sense making, problem framing, diversification, and novel and rich concepts. Usability engineering, on the other hand, focuses on the iterative improvement of a particular artifact against measurable usability objectives and qualitative understanding. Modeling can complement both. Its power lies in the abstraction and decomposition of a problem, and it is strongest at the problem-solving stage when the final artifact is described or exists. Further, modeling is not necessarily reductionist. It can take holistic, systemic, or emergent perspectives. For example, complex sociotechnical processes like word-of-mouth have been studied with models of social networks and technology adoption with dynamical models.

Modeling is key to progressing HCI. Models are not needed for the sake of scientific pretension, but rather because we simply cannot move forward without them. The three elements of HCI—the human, the computer, and interaction—are extremely complex. Our explanans must seek commensurate complexity. Models create a platform for research on complex phenomena that is transparent, collaborative, reproducible, progressive, and, yes, rigorous. Computational methods have provided seeds for a new period of intellectual growth that is within our reach. Some longstanding criticisms of modeling can now be revisited. The second sidebar collects classic criticisms and new responses.

Embarking with models is vital for HCI also because we need to address some of the most important questions of our time. Shouting from the margin will change neither AI nor the networked world. All interactive technology is run by models and we need to directly engage with them to transform them. And we must embrace math

and code to achieve that.

Getting started: We have offered a course on computational interaction at CHI since 2017 as well as a weeklong, SIGCHI-sponsored summer school. The Python notebooks from those are a good starting point: https://github.com/johnhw/chicourse_2019. Several interface metrics are collected in AIM, the Aalto Interface Metrics server. You can try it out and download it at <https://interfacemetrics.aalto.fi/>

ENDNOTES

1. Oulasvirta, A., Bi, X., Kristensson, P.-O., and Howes, A., Eds. *Computational Interaction*. Oxford Univ. Press, 2018.
2. Bailly, G., Oulasvirta, A., Kötzing, T., and Hoppe, S. Menuoptimizer: Interactive optimization of menu systems. *Proc. UIST ’13*. ACM, New York, 2013, 331–342.
3. Gajos, K. and Weld, D.S. SUPPLE: Automatically generating user interfaces. *Proc. IUI ’04*. ACM, New York, 2004, 93–100.
4. Talton, J., Yang, L., Kumar, R., Lim, M., Goodman, N., and Méch, R. Learning design patterns with Bayesian grammar induction. *Proc. UIST ’12*. ACM, New York, 2012, 63–74.
5. Kangasrääsiö, A., Athukorala, K., Howes, A., Corander, J., Kaski, S., and Oulasvirta, A. Inferring cognitive models from data using approximate Bayesian computation. *Proc. CHI ’17*. ACM, New York, 2012, 1295–1306.
6. Huberman, B.A., Pirolli, P.L., Pitkow, J.E., and Lukose, R.M. Strong regularities in world wide web surfing. *Science 280*, 5360 (1998), 95–97.
7. Jokinen, J.P., Sarcar, S., Oulasvirta, A., Silpasuwanchai, C., Wang, Z., and Ren, X. Modeling learning of new keyboard layouts. *Proc. CHI ’17*. ACM, New York, 2017, 4203–4215.
8. Masci, P., Ayoub, A., Curzon, P., Harrison, M.D., Lee, I., and Thimbleby, H. Verification of interactive software for medical devices: PCA infusion pumps and FDA regulation as an example. *Proc. EICS ’13*. ACM, New York, 2017, 81–90.
9. Hess, R.A. and Modjtahedzadeh, A. A control theoretic model of driver steering behavior. *IEEE Control Systems Magazine 10*, 5 (1990), 3–8.
10. Bachynskyi, M., Palmas, G., Oulasvirta, A., and Weinkauf, T. Informing the design of novel input methods with muscle coactivation clustering. *ACM Trans. on Computer-Human Interaction 21*, 6 (2015), 30.

Modeling is not necessarily reductionist. It can take holistic, systemic, or emergent perspectives.

Antti Oulasvirta is a cognitive scientist and an associate professor at Aalto University, where he leads the User Interfaces research group.
→ antti.oulasvirta@aalto.fi

✉ <https://userinterfaces.aalto.fi>