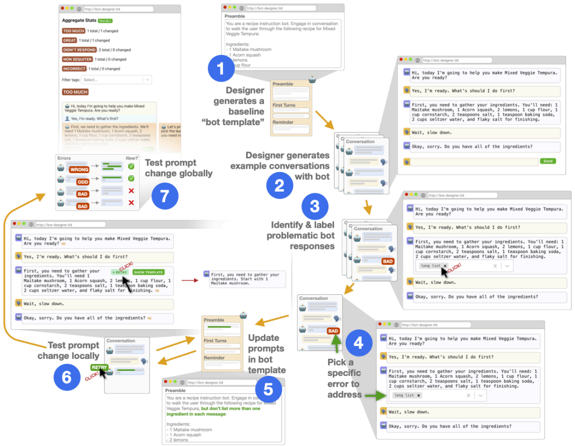
J.D. ZAMFIRESCU-PEREIRA • RESEARCH STATEMENT • Human-AI Interaction + Design

My research focuses on effective human-AI co-design. I study the boundaries of language interfaces as a medium for interacting with AI, creating systems that blend language-focused interactions with structured user interfaces that draw on different levels of abstraction. My work fits into the larger area of **Human-Computer Interaction**, I publish in top-tier venues in HCI, such as CHI, DIS, UIST, and FAccT, and I am currently supported by the Google PhD Fellowship.

I focus on language-oriented technologies, like LLMs and text-to-image models, that are powerful mediators of design processes. These technologies enable humans to describe their desires at almost any level of abstraction, from high-level goals vaguely specified ("I'd like a game to help my kid learn to read") to low-level corrections of undesired outputs ("Don't say 'I know because I've tasted it' when asked if a recipe substitution will taste good").

Natural language instruction does not remedy all problems, and, in fact, poses new challenges. Today's AI autocomplete interactions in code and emails—and the ubiquitous chatbot and prompt box interfaces imploring users to "request" anything they want—are woefully insufficient mechanisms that lead to user frustration and suboptimal outcomes. In part that is because **people ascribe humanlike capability to systems that take humanlike input, but then struggle when those systems respond in non-human ways to the breadth of that humanlike input:** In *Why Johnny Can't Prompt* [12], we show how humans interpret LLMs' humanlike outputs as though they have the same meaning they would if uttered by a human (e.g., a cooking bot saying "I know because I've tasted it") and treat LLMs as though they have preferences a human might (e.g., saying "please" to be polite, and preferring short instructions over providing extensive examples). In *Herding AI Cats* [11], we show how interactions between prompt instructions stymie fundamental engineering principles like modularity and the separation of concerns, limiting what can be done with natural language instruction alone. Together, these papers show how human intuitions, misapplied through LLMs' natural language interfaces, simultaneously lead humans astray *and* obscure these models' remarkable capabilities.

I address these challenges with systems that (a) enable large-scale exploration of AI design spaces, reducing overgeneralization risks and surfacing capabilities not intuitively explored; (b) ground interactions across abstraction levels, mitigating user frustration; and (c) structure the outputs and inputs of natural language interfaces, supporting fundamental engineering principles. For example: PAIL [9] broadens computer program design space exploration through structured design support (a, b, c); DreamSheets [1] uses spreadsheet scaffolds to create large scale small-multiples visualizations of text-to-image outputs (a, c); and BotDesigner [8] structures conversational interactions into reusable test cases (b, c). Together, these systems demonstrate ways to overcome the challenges of natural language instruction with AI.


1
Designer generates a baseline "bot template"
2
Designer generates example conversations with bot
3
Identify & label problematic bot responses
4
Pick a specific error to address
5
Update prompts in bot template
6
Test prompt change locally
7
Test prompt change globally
Aggregate Stats
TOO MUCH 1 total / 1 changed
GREAT 1 total / 1 changed
DIDN'T RESPOND 2 total / 0 changed
NON SEQUITER 1 total / 0 changed
INCORRECT 1 total / 0 changed
Errors
New?
WRONG
ODD
BAD
BAD
Preamble
First Turns
Reminder
Conversation
long list
CLICK!
RETRY
SHOW TEMPLATE
Send

Preamble: You are a recipe instruction bot. Engage in conversation to walk the user through the following recipe for Mixed Veggie Tempura, but don't list more than one ingredient in each message:

Ingredients:
- 1 Maitake mushroom
- 1 Acorn squash
- 2 lemons

Hi, today I'm going to help you make Mixed Veggie Tempura. Are you ready?

Yes, I'm ready. What's should I do first?

First, you need to gather your ingredients. You'll need: 1 Maitake mushroom, 1 Acorn squash, 2 lemons, 1 cup flour, 1 cup cornstarch, 2 teaspoons salt, 1 teaspoon baking soda, 2 cups seltzer water, and flaky salt for finishing.

Wait, slow down.

Okay, sorry. Do you have all of the ingredients?

First, you need to gather your ingredients. Start with 1 Maitake mushroom.

The BotDesigner probe enabled participants to define a chatbot's behavior using natural language prompts, then categorize problems and test changes to prompts against those categories, as in the workflow shown here.

My work includes the most downloaded CHI paper in the conference's history ([12]), my systems ([6, 10]) have been used by thousands of students in introductory computer science and data science courses, and my workflows and techniques have been adopted by multiple startups in industry ([7]).

UNDERSTANDING INTUITIONS & AFFORDANCES OF NATURAL LANGUAGE PROMPTING

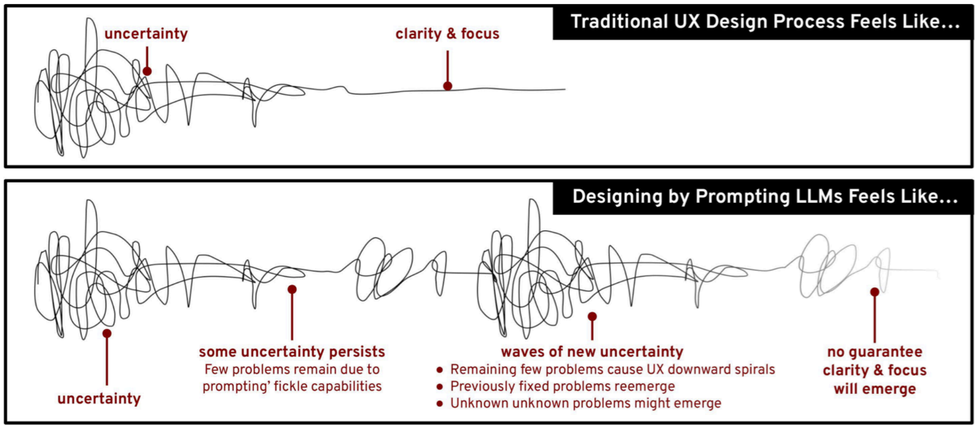
In 2022, as GPT-3 was gaining in notability, we were perhaps the first team to study how novices approach prompting LLMs via a paper called *Why Johnny Can't Prompt* [12]. Submitted for review a few months before the launch of ChatGPT, we asked participants to instruct a chatbot that walked its end-users through cooking a recipe. (Imagine an Alexa walking you through a recipe; we asked participants to "instruct" that Alexa program through prompts alone.)

Human Intuitions: We found, unsurprisingly, that natural language instructions are not a panacea for creating computing systems. Our participants relied heavily on intuitions from human-human instructional interactions—sensibly, as what other instructional interactions could they pattern match from?—and these intuitions were not only not always helpful, but also very hard to change. Participants were overly polite, and biased towards giving instructions over providing examples, even after observing repeatedly how helpful examples were—then over-generalized from single successes or failures. These results have critical implications for the design of LLM-based natural language systems, foremost among them that these systems need to disabuse their users of the notion that they behave as humans do. To the extent that every commercial computing application is racing to integrate "AI", we offer a critical insight that **people struggle to understand and direct LLMs because these natural language interfaces promise universal human-level capability across any domain—but without the ability to uphold that promise.**

These findings echo Nass *et al's* Computers are Social Actors [5] paradigm, and Ko *et al.*'s Learning Barriers [4]: early challenges can be overcome with the assumption of human-level capability, but this stalls later progress. Our results are suggestive of human use of natural language instructions *in general,* beyond LLMs—and **this work is the most-cited CHI paper of the past 3 years, and the most-downloaded paper in the history of CHI**.

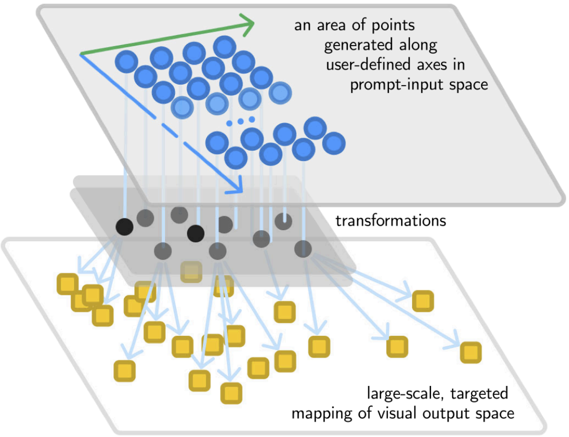
Affordances of Prompting: Experts, meanwhile, face different challenges—in *Herding AI Cats* [11], our team of chatbot, programming, and NLP experts used BotDesigner ourselves to *prompt engineer* a recipe-instruction chatbot inspired by Carla Lalli's personal style in Bon Appetit's Back-to-Back Chef, emphasizing her sense of humor, her staccato style, her frequent confirmations with guest chefs and use of vivid visual language to communicate object identities ("giant brain-looking mushroom") and intangibles ("keep adding water until it's like ooblek–you remember ooblek?").

We found that while individual behaviors were achievable, combining "subcomponent" prompts into larger prompts was quite challenging: subcomponent prompts interact unpredictably—making it hard to separate concerns, resulting in reemergent failures. We captured this frustration in our adaptation of the Design Squiggle (*see figure, right*)—while in traditional UX


Traditional UX Design Process Feels Like...
uncertainty
clarity & focus
Designing by Prompting LLMs Feels Like...
uncertainty
some uncertainty persists
Few problems remain due to prompting' fickle capabilities
waves of new uncertainty
● Remaining few problems cause UX downward spirals
● Previously fixed problems reemerge
● Unknown unknown problems might emerge
no guarantee clarity & focus will emerge

Comparing design processes across traditional UX (top) and designing-by-prompting (bottom).

design initial uncertainty often gives way to clarity and focus, when *designing by prompting*, we could never shake the uncertainty and felt no confidence about getting to a point of clarity and focus. (Memorably, eliciting humor and limiting each bot utterance to only one task seemed mutually incompatible, despite many attempts at gluing individually-functioning prompts together.)


an area of points generated along user-defined axes in prompt-input space
transformations
large-scale, targeted mapping of visual output space

An illustration of how scaffolded "axes" generated with spreadsheet formulae and LLM functions help map out the design space in DreamSheets: blue circles represent templated prompts, for example, *a {emotion} dog wearing a {hat_type}* — each variable from one axis.

LLMs appeal for chatbot design because they present as capable of handling a broad diversity of interactions *unanticipated by the designer:* adapting a recipe for specific religious restrictions, or including a child's favorite ingredient. **Designers want hallucinations, but only the right ones**—tricky because what's "right" can depend on reasoning or experience that these models don't have, and can't be provided through prompted context alone.

AI IN DESIGN: Large-scale Generation, Comparisons, and Exploring the Design Space

Design is centered around an iterative process of constructing and evaluating prototypes, enabling fast exploration of alternatives that address uncertainty about the design problem. In DreamSheets [1] and PAIL [9], we explored explicit design support for generating and comparing alternatives.

Our digital artist participants in DreamSheets identified building a mental map of models' understanding of concepts *within* prompts as critical to their processes, achieved only through generating many images. DreamSheets offers explicit cognitive support for exploring the design space of prompt inputs and image outputs for text-to-image models. This support is embedded into collaborative spreadsheet software Google Sheets, which we extended to include spreadsheet formulae for manipulating prompts: a set of LLM-based functions that turns concepts (e.g., "colors") into rows or columns ("red", "blue", "green", etc.). These columns then enable the creation of 2D small multiples views of generated images, a well-established method for comparing visual outputs, enabling our participants' rapid sensemaking through exploration within a huge design space—showing one effective way to scaffold users' understanding of how these models behave.

In PAIL [9], we studied explicit support for iterative design of computer programs, a task similarly characterized by navigating a space of alternative problem formulations and associated solutions. By default, LLMs deliver code that represents a particular point solution, obscuring the larger space of possible alternatives, some which might be preferable to the LLM's default interpretation. PAIL generates new ways to frame problems alongside alternative solutions, tracks design decisions, and identifies implicit decisions made by either the LLM or the programmer. LLM assistants can produce far more code and more alternatives than the user can process in real time, resulting in overwhelm if not well-managed—PAIL's three agents alone posed challenges for organization and information overload. Once programmers lost awareness of the (low-level) code as it evolved—even if they kept up with (high-level) design changes—regaining this awareness was cognitively demanding, showing a need for future systems like PAIL to support users in moving across different abstractions.

ACTIVE DEPLOYMENTS • CONNECTIONS TO EDUCATION & PRACTICE

My research has also had impacts in industrial practice and in CS education. I spearheaded 61A-Bot [10], an LLM-based assistant for Berkeley's largest intro CS course (CS 61A), which reduced student homework completion times by 30 min or more per assignment, a reduction 3-4 times larger than the typical variation from semester to semester. This work has also served as a testbed for understanding AI systems' influence on human learning, with clear shifts in what and how students learn: our Bot, unlike human TAs, provides multiple hints in one message, with better odds of progress [6]—but also with drawbacks: students no longer learn how to read debug messages.

My PhD student mentee's EvalGen [7] explores how humans might define desired behavior for LLMs in a way that *can be maintained over time*—using a set of assertions, co-designed with another LLM, and evaluated against a growing set of graded prompt outputs. Even discovering criteria with which to evaluate LLM outputs requires looking at a significant subset of those outputs, and our participants' early criteria would drift in hard-to-predict ways. **Since we posted our EvalGen preprint, multiple startups have already implemented our techniques in their products**.

RESEARCH AGENDA

My research goal is to build systems and use them to test theories of human and machine capability and collaboration, seeking a deeper understanding of the mechanisms underpinning design. I will continue my collaborations with artists, designers, and programmers, and expand collaborations across academic departments, especially in AI, Psychology, and Learning Sciences. Some directions I plan to pursue include:

Scaffolding Collaboration: Common Language for Grounding. Human-human natural language interaction strategies don't always work well for language models. How should humans and large models work together to construct new abstractions for building complex systems? Humans rapidly and continuously form and verify shared assumptions with other humans [2]—what grounding is needed for AI systems? My PAIL [9] work suggests two approaches: first, as LLMs build abstractions and synthesize code, they can also provide incremental updates to humans' mental models, targeted at users' existing expertise—while maintaining a model of that expertise; second, properly constructed, a complex abstraction's language (e.g., its nouns and verbs) can enable both formal and informal reasoning, supporting, e.g., formal automated test suites *and* designerly, hypothetical explorations in the space that language describes.

Understanding Programs without Code. For programming specifically, one challenge is that the code itself is not the desired design artifact—it is actually an intermediate representation that is executable by a computer in order to *produce* the desired artifact. If we rely on LLMs to synthesize that code (as in PAIL), we will need complementary tools to understand programs. What *other ways*—beyond code—are there to understand and specify programs, and what makes one or the other more effective?

Interpretable Coordination of Assemblies of Agents. As AI declines in cost, we will see many more agents assisting in design tasks, e.g., [3]. Humans have vastly different constraints—LLMs don't get bored or tired, for example, enabling new organizational forms. How should human "managers" effectively oversee and direct the goals of hundreds, thousands, or millions of agents?

REFERENCES

[1] Almeda, S.G.,† **Zamfirescu-Pereira, J.D.**, Kim, K.W., Mani Rathnam, P. and Hartmann, B. 2024. Prompting for Discovery: Flexible Sense-Making for AI Art-Making with Dreamsheets. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (2024), 1–17.

[2] Clark, H.H. and Brennan, S.E. 1991. Grounding in Communication. *Perspectives on Socially Shared Cognition*. L. Resnick, L. B, M. John, S. Teasley, and D., eds. American Psychological Association. 13–1991.

[3] Hong, S. et al. 2023. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. (Oct. 2023).

[4] Ko, A.J., Myers, B.A. and Aung, H.H. 2004. Six Learning Barriers in End-User Programming Systems. *Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing* (USA, 2004), 199–206.

[5] Nass, C., Steuer, J. and Tauber, E.R. 1994. Computers are social actors. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 1994), 72–78.

[6] Qi, L.,† **Zamfirescu-Pereira, J.D.,** Kim, T., Hartmann, B., DeNero, J. and Norouzi, N. A Knowledge-Component-Based Methodology for Evaluating AI Assistants. *Under review.*

[7] Shankar, S.,† **Zamfirescu-Pereira, J.D.,** Hartmann, B., Parameswaran, A. and Arawjo, I. 2024. Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh PA USA, Oct. 2024), 1–14.

[8] **Zamfirescu-Pereira, J.D.,** Hartmann, B. and Yang, Q. 2023. Conversation Regression Testing: A Design Technique for Prototyping Generalizable Prompt Strategies for Pre-trained Language Models. arXiv.

[9] **Zamfirescu-Pereira, J.D.,** Jun, E., Terry, M., Yang, Q. and Hartmann, B. 2024. Beyond Code Generation: LLM-supported Exploration of the Program Design Space. *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (2025). *To appear.*

[10] **Zamfirescu-Pereira, J.D.,** Qi, L., Hartmann, B., DeNero, J. and Norouzi, N. 2025. 61A Bot Report: AI Assistants in CS1 Save Students Homework Time and Reduce Demands on Staff. (Now What?). *Proceedings of the SIGCSE Technical Symposium 2025* (Feb. 2025).

[11] **Zamfirescu-Pereira, J.D.,** Wei, H., Xiao, A., Gu, K., Jung, G., Lee, M.G., Hartmann, B. and Yang, Q. 2023. Herding AI cats: Lessons from designing a chatbot by prompting GPT-3. *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (2023), 2206–2220.

[12] **Zamfirescu-Pereira, J.D.,** Wong, R.Y., Hartmann, B. and Yang, Q. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), 1–21.

† PhD or MS research mentee is first author.