Designing AI Experiences: Boundary Representations, Collaborative Processes, and Data Tools

by

Hariharan Subramonyam

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information)
in the University of Michigan
2021

Doctoral Committee:

Professor Eytan Adar, Chair
Dr. Steven Drucker
Professor Steve Oney
Professor Colleen Seifert

Hariharan Subramonyam

harihars@umich.edu

ORCID iD: 0000-0002-3450-0447

To my parents, with love.

ACKNOWLEDGMENTS

There are many things that need to happen ‘right’ to make for a positive Ph.D. experience. I am immensely grateful to everyone who believed in me and invested time, energy, and resources in making this experience exciting and fruitful. This dissertation would not have been possible without your continuous support and guidance.

First, I would like to thank Eytan Adar for being the best advisor anyone could ask for. A common response to introducing myself as your advisee is, “you chose well!” I am indebted to you for taking the risk on me. Thank you for being enthusiastic about crazy ideas and then teaching me how to turn them into meaningful research contributions. I have enjoyed the countless hours of brainstorming, and learning scholarly writing from you. As an advisor, you have always protected my interests and defended me in times of need; for that, I am grateful.

I think I got lucky to meet Colleen Seifert, a committed mentor and one of the nicest people I know. I am grateful to you for supporting my curiosity in cognitive psychology and your patience in nurturing this interest. My doctoral experience would not have been half as exciting without our collaboration. I especially enjoyed our discussions during the summers. Thank you for your endless enthusiasm, encouragement, and going above and beyond to support my Ph.D. journey.

I am fortunate to have worked with Steven Drucker during my internship at Microsoft and our subsequent collaborations and advising. Thank you for sharing my enthusiasm and supporting me in pursuing complex research problems. Your mentorship has been instrumental in building my self-confidence and maturity as a researcher. I am grateful to have you on my committee.

I am also thankful to Steve Oney for serving on my dissertation committee. I have gained inspiration from your scholarship and learned a great deal through our conversations. Thank you for your expertise and feedback on my dissertation work.

I am indebted to Priti Shah for supporting my research interest in cognitive psychology and mentoring me on numerous projects. Your excitement and energy have been a constant source of motivation. Furthermore, working with you has taught me to be a more considerate scholar.

I have been lucky to have worked with several other exceptional and compassionate scholars throughout my Ph.D. My early work with Sile O’Modhrain has greatly influenced my thinking about HCI. I am thankful to Mira Dontcheva and Wilmot Li for mentoring me during my internship at Adobe. I learned a great deal from them on effectively communicating technical HCI work. I thank Matthew Kay for serving on my Ph.D. milestone committees. I thank Paul Resnick for the opportunity to work with him as a master's student and for encouraging me to apply for a Ph.D. I am also grateful for the guidance and advice from Joyojeet Pal, Michael Nebeling, Audrey Michal, Jenn Wortman Vaughn, Hanna Wallach, Jessica Witt, Zicheng Liu, Ken Holstein, and Aysecan Boduroglu.

I would like to thank Allison Sweet, Veronica Falendino, Deborah Apsley, and Amy Eaton for accommodating my numerous paperwork needs throughout my Ph.D. I also thank Catherine Robinson for her advice and support in research administrative tasks.

I would also like to thank my many wonderful friends including Daphne Chang, Sam Carton, Jasmine Jones, Chanda Phelan, Tawfiq Ammari, Priyank Chandra, Shiqing He, Linfeng Li, Penny Triệu, Jane Im, Cristina Garbacea, Preethi Srinivas, Himabindu Lakkaraju, Teng Ye, Sungjin Nam, Gaurav Paruti, Srayan Datta, Maulishree Panday, Andrew Ku, Nick Reid, Cindy Xiong, Darshana Joshi, Ihudiya Williams, Xin Rong, Shiwali Mohan, Anusha Venkatakrishnan, Sangseok You, Chuan-Che Huang, and Joey Hsiao, amongst others for making this journey enjoyable. I look forward to our future collaborations!

I am also grateful to all my study participants for their time and feedback.

Finally, I would like to thank my parents, sister, and brother-in-law for their endless support, encouragement, and love.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Artificial Intelligence (AI) has transformed our everyday interactions with technology through automation, intelligence augmentation, and human-machine partnership. Nevertheless, we regularly encounter undesirable and often frustrating experiences due to AI. A fundamental challenge is that existing software practices for coordinating system and experience designs fall short when creating AI for diverse human needs, i.e., "human-centered AI" or HAI. "AI-first" development workflows allow engineers to first develop the AI components, and then user experience (UX) designers create end-user experiences around the AI's capabilities. Consequently, engineers encounter *end-user blindness* when making critical decisions about AI training data needs, implementation logic, behavior, and evaluation. In the conventional "UX-first" process, UX designers lack the needed technical understanding of AI capabilities (*technological blindness*) that limits their ability to shape system design from the ground up. Human-AI design guidelines have been offered to help but neither describe nor prescribe ways to bridge the gaps in needed expertise in creating HAI.

In this dissertation, I investigate collaboration approaches between designers and engineers to operationalize the vision for HAI as technology inspired by human intelligence that augments human abilities while addressing societal needs. In a series of studies combining technical HCI research with qualitative studies of AI production in practice, I contribute (1) an approach to software development that blurs rigid design-engineering boundaries, (2) a process model for co-designing AI experiences, and (3) new methods and tools to empower designers by making AI accessible to UX designers. Key findings from interviews with industry practitioners include the need for "leaky" abstractions shared between UX and AI designers. Because modular development and separation of concerns fail with HAI design, leaky abstractions afford collaboration across expertise boundaries and support human-centered design solutions through vertical prototyping and constant evaluation. Further, by observing how designers and engineers collaborate on HAI design in an in-lab study, I highlight the role of design 'probes' with user data to establish common ground between AI system and UX design specifications, providing a critical tool for shaping HAI design. Finally, I offer two design methods and tool implementations — Data-Assisted Affinity Diagramming and Model Informed Prototyping — for incorporating end-user data into HAI design.

HAI is necessarily a multidisciplinary endeavor, and human data (in multiple forms) is the backbone of AI systems. My dissertation contributions inform how stakeholders with differing expertise can collaboratively design AI experiences by reducing friction across expertise boundaries and maintaining agency within team roles. The data-driven methods and tools I created provide direct support for software teams to tackle the novel challenges of *designing with data*. Finally, this dissertation offers guidance for imagining future design tools for human-centered systems that are accessible to diverse stakeholders.

CHAPTER 1

Introduction

Artificial Intelligence (AI) is prevalent in both everyday and high-stakes software applications. AI's capabilities power a wide range of tasks, including recommending treatment options in healthcare, supporting sentencing and bail decisions, user authentication through facial-identification, email smart-reply, and many others. However, end-user experiences with AI have been a mixture of delight and dissatisfaction. When implemented correctly, AI can improve the accessibility of images on the web using object recognition or analyze complex medical data to ensure early diagnosis. Yet, in other cases, the use of AI has led to misjudgment of human behavior, bias, and harms to humans [18]. For instance, AI trained with social-media data is being used to predict whether people are "trustworthy" (mischaracterizing human behavior) [229]. A second example, facial recognition software performs poorly for people with darker skin tones [40]. Over-reliance on automation has also constrained human agency across many human-AI tasks (e.g., sharing images on social media [113]). Clearly, designing AI applications that consistently work for diverse end-users is challenging [253].

Conventional (non-AI) software programs embed logic *explicitly* by writing code; however, in AI techniques such as machine learning (ML), logic is acquired through observed patterns and behavior from data. For example, consider implementing a system that uses facial recognition to authenticate users. It is nearly impossible to express face detection and face-matching using logic programming. Instead, AI engineers write algorithms that *learn* what a face is from a large collection of images of faces (i.e., the training data) [30]. Building on this learning, engineers train ML models to analyze facial features—such as distance between eyes and shape of the chin—to match an input face to verify identity. To ensure that such a system is usable by diverse end-users and in various contexts of use, engineers must first train the face-detection model using a diverse collection of faces. Second, the features and assumptions in analysis algorithms should satisfy end-user needs, such as verifying a face when wearing glasses, under low light conditions, or wearing a face mask. Third, given inherent uncertainties in AI systems [140], user-experience (UX) designers must provide end-users with a way to recover from AI failures [10, 26]. In other words, designing AI systems around human needs will require *centering* people in defining training

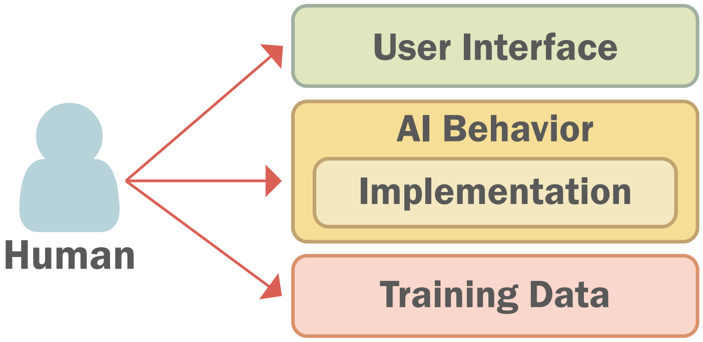


Figure 1.1: Human-Centered AI

data needs, determining the AI model behavior to be learned along with its implementation, and designing the user interface (Figure 1.1). I call this *AI Experience design* or AIX.

An objective of AIX design is to create human-centered artificial intelligence (HAI) that reflects human intelligence, is ethically aligned with human values, and is useful and usable by people (as in human-factors design) [211, 248]. To achieve HAI, researchers across HCI, AI, and the Social Sciences have offered design frameworks [212, 227], guidelines [10, 92], processes [254], and tools [93, 235] for successful HAI creation. These recommendations intersecting psychology, design, and engineering require multiple forms of expertise to implement in practice [127, 191, 196, 227]. For instance, one recommendation is that those creating HAI systems engage in reflexive criticism to uncover assumptions in designs [177], and another is to strategically build AI systems to embody desirable moral values [84]. In addition, HAI creators should consider socio-cultural perspectives such as fairness, accountability, and transparency [201] through domain experts and target users, as well as identify unintended consequences by evaluating anticipated context-specific needs [15]. In other words, creating HAI requires *multidisciplinary* expertise in constant *collaboration* to meet these defined qualities of success.

But in reality, current work-team structures and AI software development workflows make it challenging to operationalize these wide-ranging recommendations for HAI. Over the past forty years, human-centered design (as practiced) is optimized for efficiency through two distinct forms of expertise: user experience (UX) designers and software engineers [19, 95, 209]. In conventional software development, designers and engineers to work independently through *coordination* and hand-off [209] processes, such as information hiding and separation of concerns [187, 189]. That is, UX designers focus on human psychology and design by working with end-users to define system requirements. Software engineers skilled in computer programming then implement those requirements in a system [209]. However, AI systems break this mold, making such an approach impractical. As shown in Figure 1.1, human-centered design is fundamentally different for AI systems than for conventional non-AI software. For multiple reasons, AI engineers and UX designers find it challenging to incorporate human needs into AI models while working independently. Currently, we lack effective processes for designers and engineers to create Human-AI applications from the ground-up.

1.1 How is HAI design different from conventional UX design?

To understand the differences between UX for non-AI applications and HAI application design, consider designing a simple To-Do list application with a human-centered approach. For conventional software, the design process consists of information gathering, needs identification, envisioning and evaluation, and finally, requirements specification [55, 163]. The designer will first gather information from different end-users (students, IT professionals, educators, etc.) regarding how they define tasks, what types of tasks they wish to capture, and their process for managing tasks. The designer then synthesizes that information to identify a set of common challenges and needs for To-Do list management (e.g., assign end-date, update completion status, add category labels, etc.). Next, the designer generates ideas and designs several alternative interfaces to meet those needs. Knowledge about graphical user interfaces (GUI), established design guidelines, and design tools supports the design and evaluation of alternative interfaces with representative users. For instance, UI prototyping tools support complex inputs and interactions to achieve near-final user experience tests without requiring any system programming. Based on these evaluations, the designer selects a design and generates specifications, including UI designs, functional requirements, style guides, data requirements such as user name and password lengths, interaction maps, and evaluation criteria. These design specifications capture all aspects of software behavior, and can be translated into technical requirements for software code [163, 190]. As shown in Figure 1.2, UX designers first work with end-users to generate design specifications; then, engineers translate those specifications into technical requirements and, eventually, software code. In conventional (non-AI) application design, knowledge is handed across the expertise boundary of designer and engineer through the specification and hand-off of task, data, functional, and UI/UX requirements from UX designer to engineer.

Now, consider instead an AI-powered To-Do list application; such a "smart" application might automatically create task items from email content (e.g., [82]). Rather than constraining end-users on what inputs they can specify (through GUI design), the primary user experience involves predicting the task intent from naturally occurring email texts. When designing such an AI-powered application, UX designers *cannot* fully specify the design at the user-interface level on their own. As described earlier, HAI design extends beyond the user interface into the design of AI subcomponents, including AI behavior and implementation along with training data and labels. So

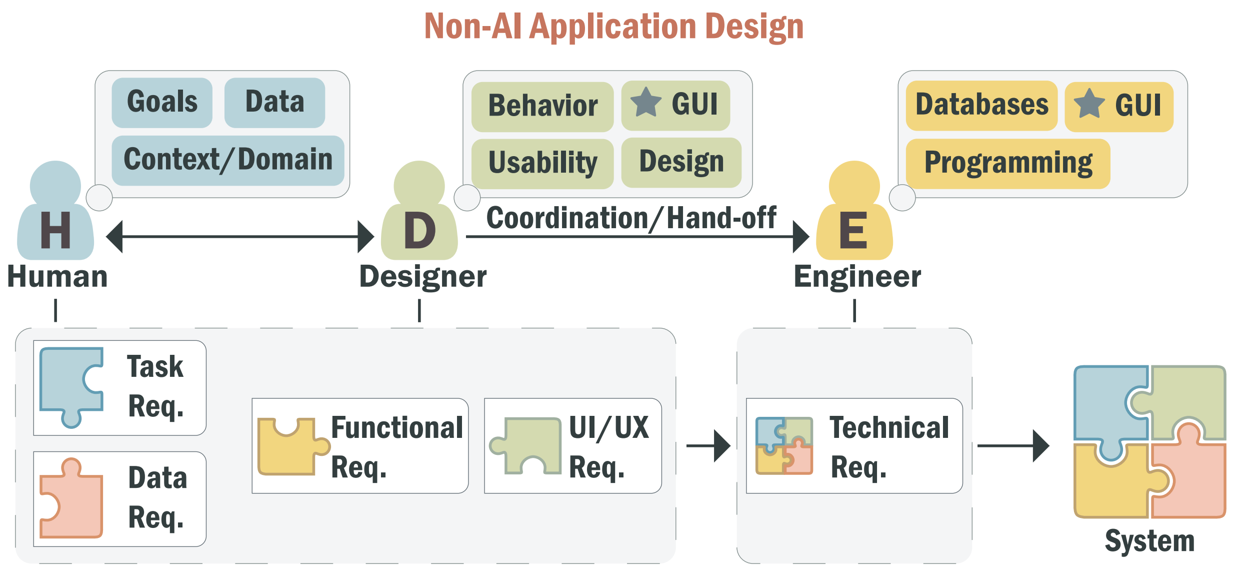


Figure 1.2: Coordination in Conventional (Non-AI) Software Design Processes

what is required for a HAI design process?

To create an AI experience with a human-centered approach, we first need to construct datasets with representative email data from diverse users covering a range of expressive email tasks. Next, we create and annotate "ground truth" data to define how users would want to generate tasks from those emails. Then, we design the AI model behavior and implementation, including assumptions, features engineering, and learnability. In determining model behavior, we need to consider how the AI experience will integrate with end-user task workflow; that is, what to automate, when to offer assistance, and when to maintain human control of tasks. We also need to account for uncertainties in AI model outputs as well as design interface adaptations for explainability, failures, feedback, and hand-off. All of this work in creating the HAI requires specifying design aspects across "all" layers of the software application stack, i.e., vertical prototyping [24]. Finally, unlike conventional applications, we cannot prototype and evaluate at the user interface level alone; instead, we need to consider fair performance across diverse users, preferably through a dis-aggregated evaluation approach [20]. As a result of this increased complexity in HAI design, the boundary between the human and interface requirements typically addressed by UX designers and the technical requirements addressed by the engineers becomes a point of friction in system creation.

1.2 Why is HAI design challenging to operationalize?

When designing AI applications, current work-team structures and expertise roles support two different approaches to HAI design. The first is a "UX first" approach similar to conventional

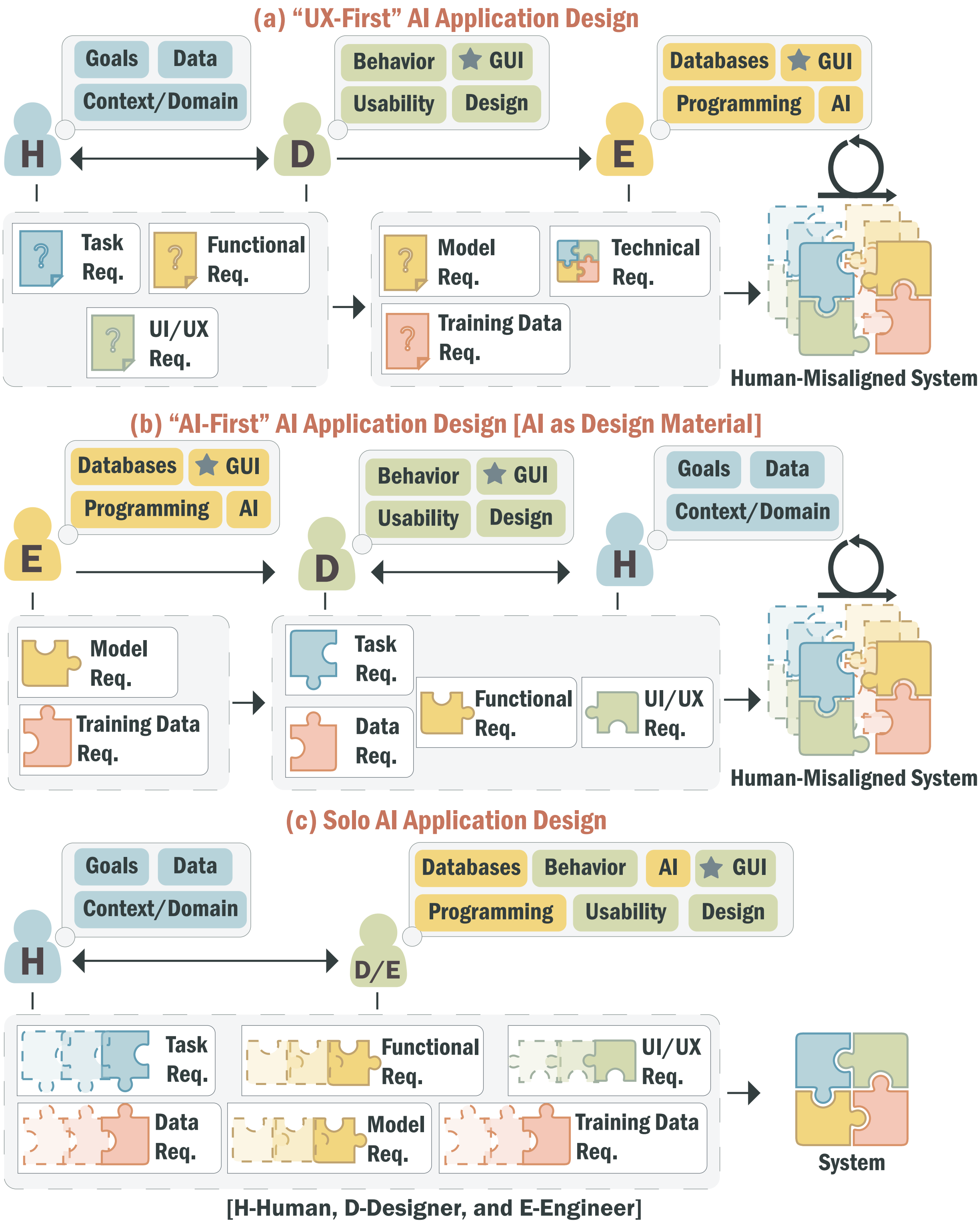


Figure 1.3: Alternative Processes for Coordinating Design and Engineering Expertise in HAI

(non-AI) software design processes. As shown in Figure 1.3 a, the "UX first" approach begins with designers working with end-users to specify UX requirements for HAI. However, for AI systems, designers (and end-users) may not know what AI can and cannot do [68, 249]. Designers experience *knowledge blindness* about AI technical capabilities, creating difficulty for them in generating task and functional specifications. For instance, they may not know how to design solutions for end-user needs using AI capabilities and [253]. Further, current design tools do not support prototyping and evaluating UX for HAI [251]. Therefore, producing interface specifications also becomes challenging. As a result, designers may make use of erroneous assumptions about AI capabilities when creating functional and interface requirements. This *premature specification* of UX design without considering AI capabilities and model uncertainties is then handed off to the engineer. These under-informed design specifications complicate the engineer's task of building the HAI system. Engineers may need to fill in missing details when defining the AI model and training data requirements. The likely result of premature design specification and jerry-rigged technical requirements will produce HAI systems that do not align well with human needs.

The second approach to HAI design is an "AI-first" approach in which AI engineers first develop AI capabilities for a desired task (Figure 1.3 b). Then, the AI model is handed off to designers create the user experience and provide necessary adaptations to support end-users. In this "AI-first" approach, designers must treat AI as a technology design "material" and explore its capabilities and limitations in order to design user experiences [117, 250]. The challenge with this approach is that engineers experience knowledge blindness about end-users when creating model and training data requirements. Developing the AI first also leads to *premature specification* of the HAI system; and as the designer identify necessary changes to align the AI's properties to human needs, require rework will be costly. Most importantly, designers lack the necessary methods or tools to work with AI as a design material; for example, they need to know what AI capabilities are possible and how the AI can be adapted to produce desired behavior. Consequently, this "AI-first" approach also results in HAI systems that do not align with human needs. A third approach combines both AI and UX design expertise within a solo designer (Figure 1.3 c). In the case of technical HCI researchers or independent application developers, trans-disciplinary skills allow them to iteratively specify both AI and UX requirements while eliminating the problem of knowledge blindness. In my own work building interactive intelligent systems [223, 225], I have found that that creating HAI requires iterating between designing functional (form-giving) prototypes and 'repairing' the AI material to fit the design (indicated in the figure by dotted puzzle pieces). However, individuals with such expertise are atypical within industry practice. An alternative process is needed to support design and software expertise during the creation of HAI. In this dissertation, I take a human-centered design perspective to explore how designers and engineering practitioners can work together to achieve the vision for AIX. Specifically, I investigate how designers and engineers might collaboratively design both the AI components and user experience by considering human needs. My thesis is that:

> ***Unlike conventional software practices that favor clear separation of concerns, creating AI experiences with designers and engineering practitioners benefit from (1) "leaky" abstractions to share information across different layers of the application, and (2) delayed specifications through vertical prototyping, and (3) constant evaluation using data tools.***

1.3 Contributions

This dissertation identifies problems in AIX design and ways to bridge *knowledge boundaries* between designers and engineers as well as overcome the problem of *premature specifications* for AI and UX components. As shown in Figure 1.4, an alternative to AI-first or UX-first design workflows, I identify a co-design approach in which designers and engineers *collaborate* in designing both the AI and UX components. To support collaboration, I conceptualize a software workflow that blurs abstraction boundaries between AI and UI, put forth a process model for co-design, and develop data-driven design methods and tools for AIX design. My contributions are:

1.3.1 Leaky Abstractions and Delayed Specifications for AIX Design

Through interviews with UX and AI practitioners, I identified creative workarounds and boundary representations used to operationalize HAI guidelines in practice (Chapter 3). In conventional software development workflows, teams coordinate primarily through interface-level abstractions such as APIs. For instance, designers do not expose details from interviews and observations that informed interface designs with engineers. Similarly, engineers do not share program-level details (program logic, rules, and assumptions) with designers. Instead, teams favor strict abstractions and clear boundaries. However, teams disregarded conventional software abstractions in AIX design and exposed low-level design and implementation details across knowledge boundaries–called leaky abstractions. Through abstraction leaks at the boundaries, teams align AI implementation decisions with user experience designs and verify that human needs are reflected in AI subcomponents and training data decisions. Further, to overcome the friction from premature specifications, successful teams delay committing to design solutions until later stages of the design process. Instead, designers and engineers share emerging design specifications through leaky abstractions and constantly evaluate and revise their designs based on updated understandings of human needs and AI constraints.

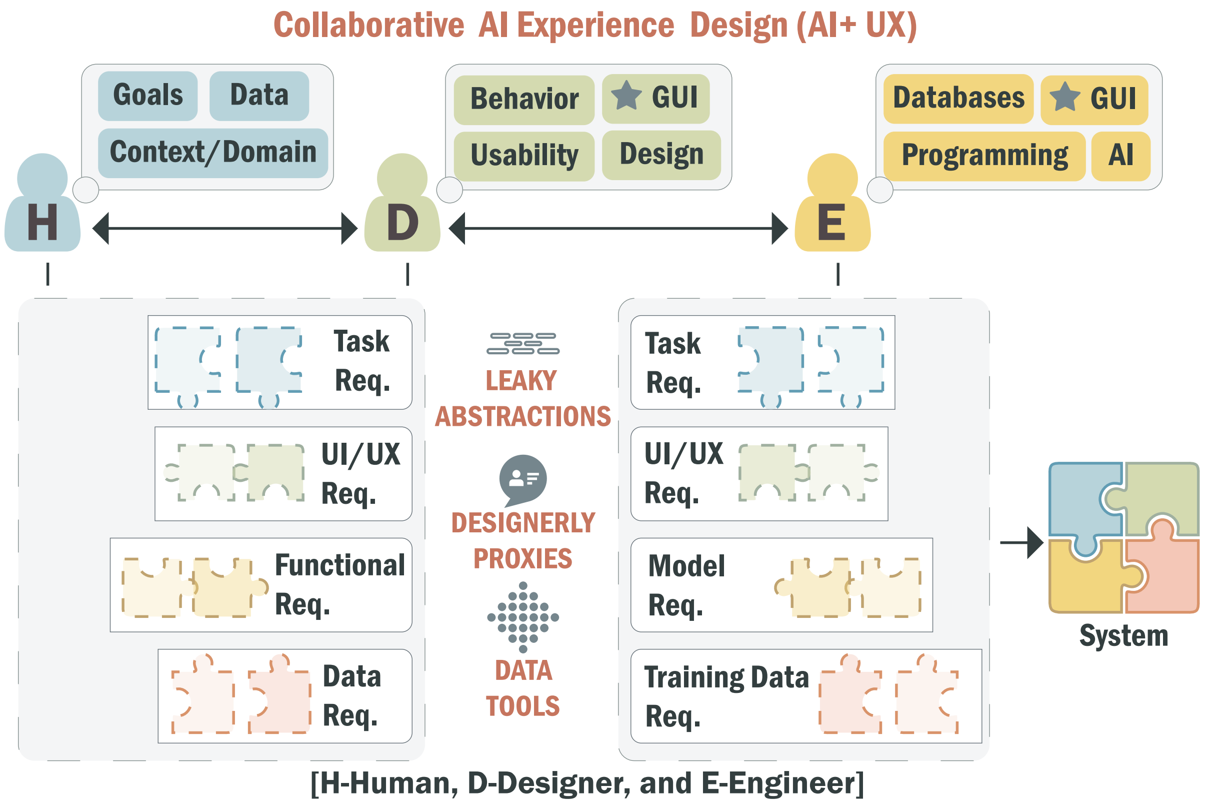


Figure 1.4: Collaborative Approach to HAI Design

1.3.2 A Process Model for Co-Creating AIX

To follow up on the findings regarding leaky abstractions, I conducted an in-lab design study to observe how designers and engineers collaborated on a given AIX design problem (Chapter 4). Based on observations, I developed a process model for co-creating AI experiences from the ground up. In this process model, the AI and UX components are designed collaboratively in parallel. I found that design "probes" with user data are a valuable tool in defining AIX design. Through data probes, designers construct designerly representations of the envisioned AI experiences (i.e., their designerly proxies) to identify desirable AI characteristics. Data probes facilitate divergent design thinking, AI behavior testing, and AIX design validation. The co-creation approach also shifts engineers' mindsets towards more proactive engagement through accessible user-data proxies. The process model distributes agency between designers and engineers and lays the groundwork for aligning AI's form and function during the early stages of design.

1.3.3 Data Tools for AIX Design

To operationalize the process model, designers need support in constructing designerly proxies using data probes. I created novel tools and techniques for two critical steps in the process model: (1) creating diverse data-personas and (2) prototyping AIX interfaces. In my interview study, designers described constructing data personas by combining qualitative insights from user research with quantitative data from end-users to define training data needs. Through data personas, designers defined training data needs by considering diverse users and their contexts of use. In contrast, existing analysis tools (such as affinity diagrams) lack support for working with quantitative data. To address this issue, I developed a novel method called *Data-Assisted Affinity Diagramming (DAAD)* and an implementation called *Affinity Lens* (Chapter 5). Using computer vision and augmented reality, Affinity Lens overlays quantitative insights on top of physical sticky notes. Affinity Lens implements several AR overlays (called lenses) to help designers cluster information and summarize insights to create nuanced data personas for AI application design. Affinity Lens supports easy switching between qualitative and quantitative ”views” of data without surrendering the lightweight benefits of existing Affinity Diagram practices.

In addition, when prototyping AIX interfaces, designers need to consider how AI features might respond to diverse end-user inputs. Current prototyping tools make it challenging for designers to work with AI subsystems during the design process. Designers need multiple tools to explore ML models, understand model performance across diverse inputs, and align AI behavior with interface choices. This adds friction to the practice of rapid and iterative prototyping. To support designers, in Chapter 6, I devised a novel prototyping workflow called *Model-Informed Prototyping (MIP).* I developed a tool called *ProtoAI* that implements MIP to combine AI feature exploration within UX prototyping tasks. ProtoAI allows designers to directly incorporate AI outputs into interface design, evaluate design alternatives across diverse inputs to the AI, and iteratively revise their design by analyzing AI breakdowns. MIP provides a foundational approach to open the ”black-box” of AI shown in existing design tools by making AI components accessible to designers.

1.4 A Note on Authorship

I am the primary author of the research reported in this dissertation. However, this work is done in collaboration with my advisor Eytan Adar from UMSI, my mentor Colleen Seifert from the Department of Psychology, my mentor Steven Drucker from Microsoft Research, and Jane Im, my colleague at UMSI. In Chapter 3, Jane Im supported me in conducting the interviews as a note-taker and contributed to qualitative data analysis. The findings and discussion are heavily informed by my two-year-long conversation with Eytan Adar and Colleen Seifert. Chapter 4 on the Process Model is published at DIS'21 and is co-authored with Eytan Adar and Colleen Seifert. Chapter 5 is inspired by work on mixed-reality visualizations with Steven Drucker at Microsoft Research. Affinity Lens is published at CHI'19 and is co-authored with Steven Drucker and Eytan Adar. Chapter 6 on Model Informed Prototyping is co-authored with Colleen Seifert and Eytan Adar and is published at IUI'21. In chapters 3-6, I use the first-person plural (we/our) to indicate co-authorship.

CHAPTER 2

Related Work

Human-Centered AI reframes the rather statistical (objective) view of AI as 'algorithms to model data' to AI as technology that *"augments the abilities of, addresses the societal needs of, and draws inspiration from human beings [167]."* Research in HCI and AI communities has characterized and detailed domain-specific viewpoints [8, 40, 66, 219], identified challenges [68, 161, 253], and put forth requirements and strategies [10, 14, 23] to operationalize the vision for HAI. Designers of algorithms (i.e., AI designers) should incorporate human and social interpretations for AI design through (1) theoretical understanding of human behavior, (2) participatory strategies to verify algorithmic assumptions, and (3) speculative and counterfactual approaches to ensure value from AI system in use [23]. By examining the complex dependencies across different components in the machine learning lifecycle, prior work has laid out desiderata for ML tasks, including the need for contextually relevant and balanced datasets and comprehensive and comprehensible verification of ML models [7, 14]. Further, designers of human-AI interfaces (UX designers) must create optimal AI experiences that balance automation and human agency by accounting for AI uncertainties and failures [8, 10, 92]. Combining these rationalistic and design goals for HAI requires multidisciplinary collaboration [15]. In this chapter, I synthesize what is known about software development process, expertise and design workflows, and boundary representations to identify challenges to HAI design and formulate the high level research questions for my dissertation.

2.1 Modular Software Development and Challenges for HAI

Human-centered software development (HCSD) is a complex problem requiring knowledge and expertise beyond what any single person can possess. When multiple individuals are involved (UX designers, software engineers, and database experts, etc.), the preferred approach is to decompose the system into modules and tasks that can be carried out relatively independently by different people [2, 209]. Often, system modules and work-team structures observe a homomorphic relation [51]. For instance, UX professionals create the user interface, and engineers implement the underlying functionality. Further, to reduce dependencies between tasks, teams agree on the definition of the module's outward-facing *interface* while the *implementation* details are abstracted from others (i.e., the principle of information hiding) [189].

Application Programming Interfaces (API) are a widely used approach for information hiding. However, good APIs are hard to specify [112]. Human-centered API implementation consists of four stages, including (1) API design, (2) API implementation, (3) API execution semantics and use cases, and (4) API optimization [197]. Each stage requires validation with end-users [197]. While APIs should be defined through collaboration between multidisciplinary teams [35, 36], knowledge and process barriers between UX and engineers may it challenging to work together [180]. Instead, in HCSD, designers take a"UX first" approach to specify the API through a 'user interface' design process [216]. Here, the user interface can be thought of as the highest level module for end-users to invoke. Designers can map end-user needs into interface design specifications. Engineers who also understand the language of the user interface can translate interface representation into implementation [209]. Anything that is expressed as interface requirements can be programmed. In other words, the user interface acts as a natural 'seam' for designers and engineers to coordinate.

However, such interface level abstractions quickly breakdown when designing AI-powered applications. First, it can be challenging to enforce strict abstractions by specifying concrete APIs [207]. In fact, ML is beneficial in cases in which behavior cannot be explicitly specified through software logic. In addition to user interface APIs, human-centered design for machine learning applications also includes estimator APIs for building and fitting models, predictor APIs for determining inputs and outputs for making predictions, and transformer APIs for defining data representation needs and converting data to train models [39]. Second, the contract nature of APIs hides implementation details that are necessary for designing AI adaptations, such as explainability and feedback [10, 45]. With ML, designer and engineers needs to bridge abstraction levels along a *part-whole hierarchy* to center people in design of AI sub-components, and within an *implementation hierarchy* to offer interface adaptations to AI uncertainties [236]. Third, APIs favor independence and limit collaboration between designers and engineers during recomposition (the work necessary to build a system from its pieces) [60]. Boundary erosion in ML often leads to technical debts, including entanglement, hidden feedback loops, undeclared consumers, data dependencies, and configuration issues. [207]. Clean handoffs between teams in ML development are difficult.

RQ1: How should we coordinate specification and implementation with designers and engineers in HAI Design?

2.2 Knowledge Barriers and Design Challenges

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 [62]. However, current UX designers are not trained in most of these aspects of HAI systems. First, UX designers lack the expertise to generate design ideas for incorporating AI in human tasks [68, 253]. They misunderstand the capabilities of ML models and propose designs that can be difficult to implement [139]. Second, given that AI takes a long time to build [252], rapid prototyping with ML through a "fail fast, fail often" approach characteristic of UX design is challenging for HAI [251]. Moreover, AI requires vertical end-to-end prototyping to identify uncertainties and edge cases and create UI adaptations [16, 24, 52]. But black-box views of ML make it difficult for designers to understand, design, and evaluate with AI [110, 111]. Third, UX processes favor creativity and imagination of desired futures, which contradicts AI's emphasis on specificity and accuracy [249]. This introduces friction into the design thinking process for HAI systems.

Similarly, engineers focused on algorithms and techniques fail to consider human perspectives during initial experimentation and AI prototyping processes [114, 161]. 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 [7, 109, 203]. But expertise in HCI and involvement in exploring human needs is lacking in engineering training. Engineers who are ML novices were shown to experience breakdowns in early-stage software development due to lack of specialized design schemas, insufficient understanding of the design process, and sub-optimal design solutions [41, 96]. Consequently, even when designers suggest modifications for better human-centered experience design, model and data changes to the AI may be challenging to execute. In AI techniques such as deep learning, it can be challenging to identify specific functional areas so as to address human user issues [13]. Further, by focusing on creating the algorithm, engineers often fail to consider the AI experience as a whole, and their involvement in UX design tapers [68, 89]. AI and UX practitioners can benefit from a symbiotic relationship [46]. HCI perspectives about the user interface can improve AI through better quality feedback on performance [205]. Presentation of AI outputs can impact the end-users subjective perception of errors and how they adjust their expectations about AI [144].

RQ2: How should designers and engineers collaboratively design both the AI and UX through a human-centered approach?

2.3 Boundary Representations for Collaboration

In complex domains such as HAI, knowledge differences or "symmetry of ignorance" between HCI and AI professionals would ideally be addressed through collaboration and social creativity [81]. Prior work on software collaboration has identified three types of knowledge boundaries, including (1) assembling–how information should be structured, (2) designing–how information artifacts are designed, and (3) intended user interaction–how users interact with designed information [243]. The goal for collaboration is to bridge these knowledge boundaries to acquire *common ground* for interaction [217]. Common ground in collaborative work includes *content* common ground and *process* common ground [48, 169]. In HAI, the content common ground is the *data* which forms the backbone of machine learning (AI) applications, and the process entails the *design processes* in creating both the AI and the UX. Further, these knowledge boundaries can be bridged by either *converging* content and process knowledge bases through elaboration, discussion, and negotiation of dependencies across boundaries (i.e., traversing knowledge boundaries) or through knowledge *transcendence* by integrating just the necessary information for collaboration through co-created scaffolds (i.e., parallel representations) and dialog around scaffolds [164].

Boundary Objects [153, 218] such as external representations play a critical role in bridging knowledge boundaries by supporting information sharing, interpretation, negotiation, and co-design. In collaborative design, these representations also include *epistemic* objects such as artifacts of design-pursuit characterized by lack of incompleteness and *technical* objects including design tools that support the process of design inquiry [76]. Epistemic objects or intermediate-design representations focus on different aspects of design and constantly change between the early and late stages of the design process through design evolution and stabilization [236]. Intermediate design representations can be categorized into artifacts between designers and objects of their activities and those between multiple designers (i.e., inter-designer compatible representations). Further, in contexts in which the boundaries are blurry and non-standard, material artifacts support the process of characterizing boundaries and collaboration (i.e., boundary negotiation artifacts) [151]. These artifacts consist of (1) self-explanation artifacts for learning, recording, organizing, remembering, and reflecting, (2) inclusion artifacts for proposing new concepts, (3) compilation artifacts to coordinate and align knowledge, (4) structuring artifacts to establish principles at the boundaries, and (5) borrowing artifacts that are repurposed in unanticipated ways across communities to augment understanding [150]. The eventual representation created by the designers through collaboration is the artifact's *specifications* encapsulating the *what*—the artifact product itself, the *how*—the procedure by which it should be implemented, and the *why* (design rationale)—the reason why the design should be as it is [236].

In software development, prototypes are commonly used as boundary objects [126]. During the process of innovation and generative thinking, prototypes serve to bind needs and technical information and can include design prototyping, prototypes for verification, prototypes for exhibition, etc. [115, 237]. In HAI, prototypes should promote agreement in defining task specifications, communicating states of design, identifying references of central notions, and negotiating weights of criteria and constraints [236]. Further, co-creating is vital to bridging knowledge boundaries [125, 153, 235]. But given the already-existing collaboration challenges, boundary object creation (prototyping) mirrors these socio-technical conflicts [141]. Therefore, we need new prototyping approaches for defining specifications that include process, content, structure, and form [155]. Further, boundary prototypes should embody a means-ends hierarchy for envisioning HAI in which each level specifies the what, the how of the level below, and the why of the level above [155]. Prior work has identified characteristics of effective boundary prototypes, including interpretive flexibility, plasticity [141], and translucency [45, 73]. These characteristics support (1) establishing a shared syntax language, (2) concrete means to learn about differences and dependencies, and (3) joint knowledge transformation without causing information overload [42].

RQ3: How should HCI and AI professionals collaboratively prototype HAI artifacts to bridge knowledge boundaries?

2.4 Data in HAI Design

Human-centered design is driven by information gathered from end-users; that is, people who use the completed system to meet their own task goals [163, 236]. Prior research has identified parallels and distinguishing characteristics between design exploration in HCI and data exploration in machine learning. In HCI, end-user data, including interview data, surveys, observational notes, and even system logs, are analyzed to identify user needs and system requirements. UX designers use this data to form conceptual models to support the design process and represent the needs, expectations, and values of end-users. In most cases, data in UX design is synthesized into higher-order information representations through convergent design processes [55]. These data abstractions are explicit, providing ways to share conclusions about design goals and values for user experience. Designers can support human-centered data science through hypothesis generation and data inquiry, including human-centered data labeling and validating data for representativeness [147, 179].

In contrast, AI systems take in raw input data as the backbone of machine learning applications driving the resulting end-user experience. For instance, ML workflows involve data in divergent tasks to explore answers and acquire data, and in convergent processes for filtering, aggregating, and reaching conclusions from data [85, 146]. Data science processes tend to be objective (mechanical) and less organic [91]. But statistical models used in ML do not encode or capture similar values to those embodied in HCI data abstractions; in fact, they are explicitly nonobservable. Further, the process by which ML algorithms learn from datasets to create personalized experiences for individual users is largely based on use contexts, a UX component [43]. While the central role of data in both could potentially bring together HCI and AI processes, current design practices and tools make it difficult to connect the roles of data in UX design with the data roles in ML processes [250]. Furthermore, ML practitioners tend to overlook human-centered needs in the creation of training data [204]. In this dissertation, I explore the potential role of end-user data in supporting the collaborative design of AIX.

CHAPTER 3

Understanding Current Practices for Collaboration in HAI

This chapter reports findings from our investigation on how designers and engineers bridge conventional boundaries and development workflows to co-design HAI applications. First, we collected and analyzed a total of 280 HAI design guidelines from across different industry sources. From this analysis, we derived a component model for designing HAI applications that span across data, ML model, user interface, and end-user mental models (see Figure 6.2). Using this model as a guide, we interviewed 21 industry practitioners (UX designers, AI engineers, data scientists, and product managers) across 14 different organizations to understand their current practices for designing and developing HAI. Based on these interviews, we identified sources of friction in a 'human-centered' AI development process and how practitioners currently bridge the design-engineering boundary. Our findings show that human-AI applications rarely begin with end-user needs. Lack of AI understanding makes it challenging for designers to define AI experiences upfront. Instead, designers work to shape user experiences around AI innovations. We also find that as opposed to rigid boundaries and information hiding, "leaky" abstractions across boundaries facilitate the collaborative design of Human-AI applications.

3.1 An Analysis of Human-AI Design Guidelines

Several software organizations, including Microsoft [6], Apple [128], and Google [92] have published design guidelines for HAI applications. Ideally, these guidelines represent best-practice advice derived from successful practices within companies. They offer a starting point to explore how collaborative teams might implement human-AI applications. For the analysis, we collected a total of 280 guidelines from various industry resources (249 after removing or combining similar guidelines). We then conducted an affinity diagramming exercise [208] to identify key topics within the guidelines (Figure 3.1). Based on the topic hierarchies in the affinity clusters, we developed a component model view of human-AI design guidelines. As shown in Figure 6.2, the model

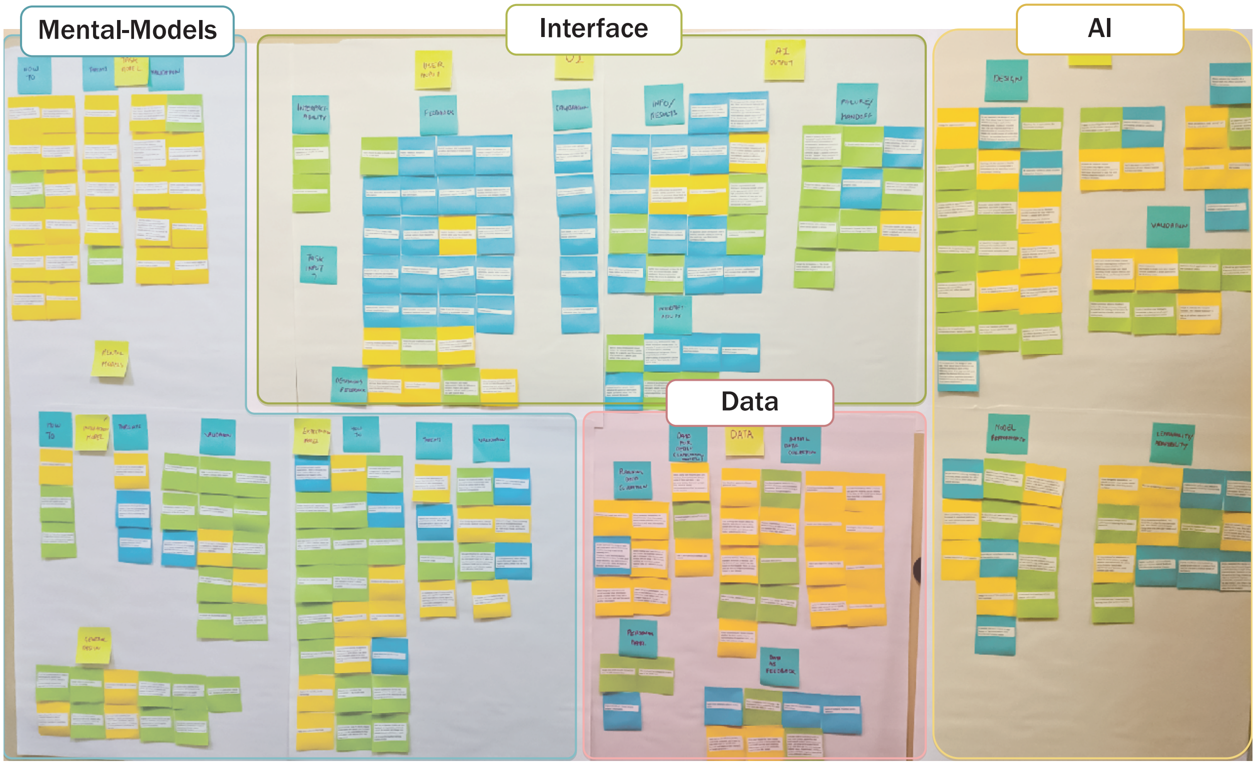


Figure 3.1: Affinity Diagram Analysis of Human-AI Design Guidelines

consists of four main components, including (1) human mental models, (2) user interface, (3) AI models, and (4) training data.

Further, while the guidelines make recommendations about the design of the components, they do not describe (or prescribe) *how* design, and engineering teams are to collaborate to put the guidelines into practice. For example, guidelines recommend that we "Align feedback with model improvement [92]" and helps users "understand how often the AI may make mistakes [6]." However, the guidelines do not indicate how teams collaborate to align AI capabilities and uncertainties with end-user experiences or communicate about those uncertainties with end-users. To find out what practices are currently in place, we used the affinity cluster to generate an initial set of questions for an interview study. To create the questions, we identified the key *nouns* (e.g., data, human-needs) and *verbs* (e.g., collection, aligning) from the guidelines in each cluster. We then translated them into questions about *who* implemented the guidelines, and *how?* Here, we summarize both the guidelines and questions for individual components in the model.

3.1.1 A Component-Model Representation of HAI Guidelines

The model consists of four main components, including (1) human mental-models, (2) user interface, (3) AI models, and (4) training data.

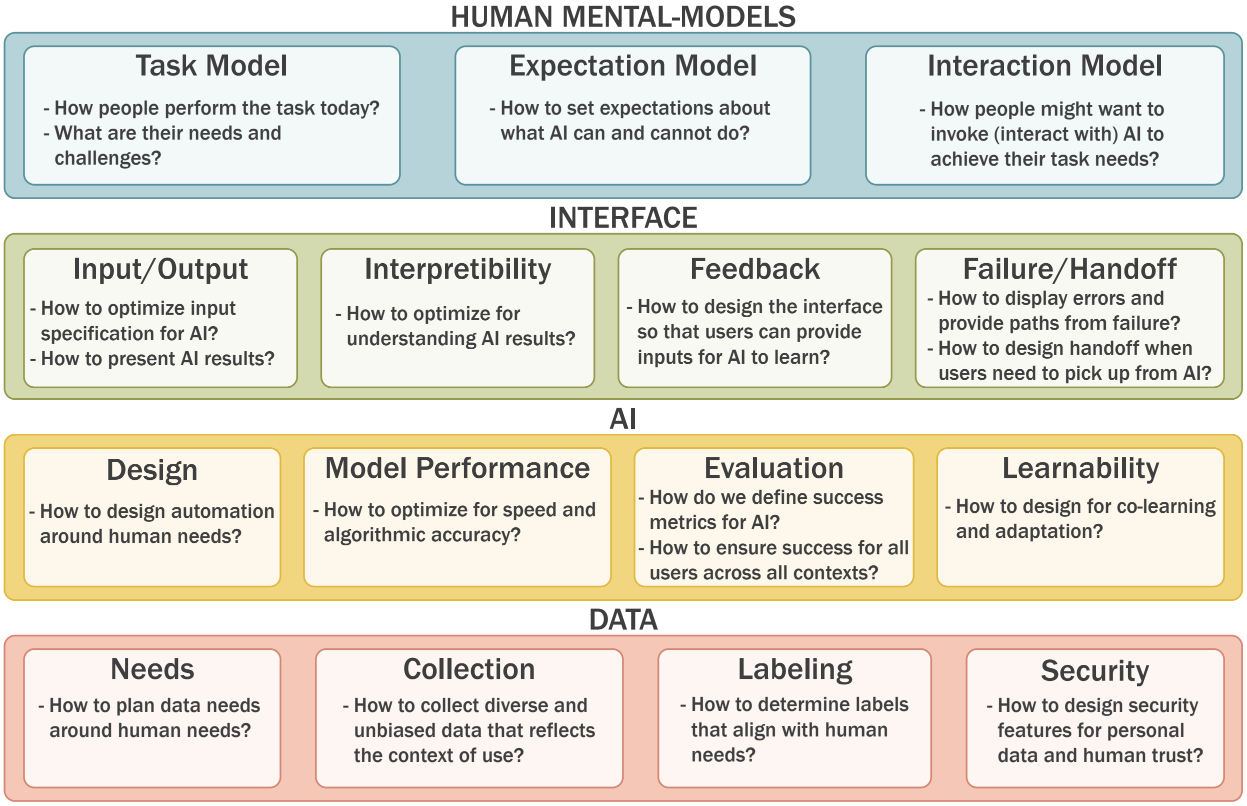

HUMAN MENTAL-MODELS
Task Model
- How people perform the task today?
- What are their needs and challenges?
Expectation Model
- How to set expectations about what AI can and cannot do?
Interaction Model
- How people might want to invoke (interact with) AI to achieve their task needs?
INTERFACE
Input/Output
- How to optimize input specification for AI?
- How to present AI results?
Interpretibility
- How to optimize for understanding AI results?
Feedback
- How to design the interface so that users can provide inputs for AI to learn?
Failure/Handoff
- How to display errors and provide paths from failure?
- How to design handoff when users need to pick up from AI?
AI
Design
- How to design automation around human needs?
Model Performance
- How to optimize for speed and algorithmic accuracy?
Evaluation
- How do we define success metrics for AI?
- How to ensure success for all users across all contexts?
Learnability
- How to design for co-learning and adaptation?
DATA
Needs
- How to plan data needs around human needs?
Collection
- How to collect diverse and unbiased data that reflects the context of use?
Labeling
- How to determine labels that align with human needs?
Security
- How to design security features for personal data and human trust?

Figure 3.2: Component Model Representation of Human-AI Guidelines

3.1.1.1 Human mental-models:

One set of guidelines focuses on understanding end-user needs in order to design AI features. Specifically, they 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 AI should do, and setting expectations for people about what AI can do, which we call the *expectation model*, and (3) identifying the right kind of AI interaction experience given people's situational context; namely, the *interaction model*.

Guidelines about the task model include identifying differences between how experts and non-expert users might perform a task, mapping their task workflows, gathering information about the diversity of users, use cases, and environment of use, and eliciting values associated with the task, such as enjoyment and human-agency. For instance, one of the guidelines about AI design recommends that we should identify opportunities for AI by understanding existing task workflow: *"mapping the existing workflow for accomplishing a task can be a great way to find opportunities for AI to improve the experience. [92]"* Further these guidelines recommend considering the overall task experience with AI to assess success: *"Successful augmentation is often measured by the following: Increased user enjoyment of a task, Higher levels of user control over automa-*

tion Greater user responsibility and fulfillment, Increased ability for user to scale their efforts, Increased creativity [92]" Our interview questions about the task model focus on how UX professionals gather, synthesize, and communicate their understanding of task-model for designing AI models.

Guidelines about the expectation model recommend assessing end-user expectations about AI behavior and the type of assumptions they expect AI to make about them, finding intersections between user needs and AI strengths, and communicating AI capabilities and limitations to users to help them calibrate their trust and expectations about AI. One of the guidelines recommend that we "help the user understand what the AI system is capable of doing [6]" However, it is unclear from the guidelines how UX researchers acquire understanding about AI behavior and its capabilities and how they formulate expectation models with end-users. Our interview questions target both of these concerns. Guidelines about interaction models suggest finding the right balance between control and automation, determining when to automate and when to offer assistance, and assessing proactive and reactive interaction experiences based on context. For example, guidelines recommend that we *"make it easy (for end-users) to dismiss or ignore undesired AI system services [6]."* The questions about the interaction model focus on how designers formulate the type of end-user experience by considering AI affordances.

3.1.1.2 User Interface:

User interface guidelines target the software and hardware interface between end-users and AI. They make recommendations about lowering the gulf of execution and evaluation by designing for (1) end-user inputs and AI outputs, (2) explainability, (3) feedback, and (4) failures and handoffs [183]. For input specification, the guidelines prescribe minimizing human effort, using the correct interactions, and demonstrating how to get the best results to end-users based on their inputs. The guidelines suggest visually differentiating AI-generated content, offering diverse alternatives under uncertainty, and making connections between end-user actions and the presentation of AI outputs. Further guidelines also recommend that HAI interfaces explain the rationale behind specific outputs such as confidence scores, categorized output displays, and interactive 'why' messages explaining algorithmic behavior. For example, when guidelines recommend that when presenting uncertain AI outputs we should *"prefer diverse options and when possible, balance the accuracy of a response with the diversity of multiple options [128]."* To accomplish these recommendations, designers and engineers must collaboratively specify the application programming interface (API) for AI features. The interview questions seek to understand the API design and negotiation process occurring between designers and engineers.

Further, the HAI guidelines make recommendations for dynamic AI experiences that involve learning from end-user feedback. For instance, some guidelines recommend that designers collect feedback to improve the model performance by mapping feedback to data labels and model parameters. This includes implicit feedback from user interactions as well as explicit, intentional feedback from end-users. The guidelines also suggest conveying how end-user actions will impact future AI behavior and allowing users to calibrate what AI has learned from them. Implementing these guidelines requires that UX designers understand low-level details about the AI model itself. Our interview questions about feedback probe how designers and engineers negotiate the feedback needed for AI model improvement. A final aspect of user interface design offered in the guidelines involves communicating AI failures by facilitating hand-off between automation and manual execution of tasks. For instance, the guidelines recommend designing the UI such that end-users are inclined to be forgiving in the case of failure and providing easy ways for users to edit, refine, or recover from AI failure. We focus the interview questions on how designers understand AI failures and how they collaboratively design AI failure experiences for end-users.

3.1.1.3 AI Model:

HAI guidelines for AI models focus on designing AI features in a 'human-centered' manner. This includes (1) designing the AI-based on end-user mental models; (2) designing for co-learning and adaptation; (3) defining model performance in a human-centered way; and (4) evaluating AI across a range of use scenarios. Regarding the AI model design, guidelines emphasize that the design should reflect information, goals, and constraints that human decision-makers weigh when making decisions, avoid unwanted biases and unfair stereotypes, and evaluate the impact of AI "getting it wrong [6]." These AI model guidelines mirror the task and expectation model subcomponents of HAI design. The interview questions thus focus on how designers communicate and collaborate with engineers in designing the AI model. Guidelines for AI feedback include adapting AI behavior based on usage, limiting disruptive changes when updating and adapting AI behavior, and designing for continuous feedback from end-users. These guidelines require balancing AI model needs with the end-user's experience of feedback. Our interview questions regarding feedback focus on (1) translating AI model learning needs into interface design and (2) designing the user experience for evolving (changing) AI behavior over time.

Further HAI guidelines recommend defining model performance in a human-centered way. This includes considering human values when weighing the cost of false positives and negatives, ensuring that model metrics such as accuracy are appropriate to the context and goals of the overall system, and making conscious trade-offs between precision and trade-offs. For instance, guidelines recommend that *"while all errors are equal to an ML system, not all errors are equal to all people. You will need to make conscious trade-offs between the precision and recall of the system [92]."* Similarly, guidelines about evaluating AI features recommend assessing whether model objectives provide a good experience for all users, evaluating for safety and whether the AI design performs under the “realities of the environment in which it to be used. Based on these guidelines, the iterative nature of AI evaluation is clear. However, it remains unclear how designers and engineers work together to define model performance metrics or how they evaluate the model behavior.

3.1.1.4 Training Data:

HAI guidelines recommend that the data needs for training the AI model are based on human needs. This 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 that data needs are aligned with the task model by asking what information a human will use to perform the task on their own [92]. 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. Implementing these guidelines requires that designers and engineers work together to scope data needs based on AI model needs and human task and expectation models.

For labeling data, HAI guidelines focus on using the right labels; that is, data labels must reflect the diversity and cultural context of the people who will use it. When creating labels, guidelines recommend ensuring rater pool diversity, rater context and incentives, and evaluating rater tools for biases. Lastly, collecting and labeling data require committing to fairness and taking steps to mitigate problematic biases in the dataset, protecting identifiable information about people, and being transparent about what data is collected and used to train the AI model features.

While the HAI guidelines make recommendations about designing AI applications in a human-centered manner, they leave a number of open questions about how to put the guidelines into practice. We investigate these questions through an interview study with industry experts.

3.2 Method

3.2.1 Procedure

We conducted semi-structured interviews with 21 industry professionals from 14 different organizations of differing sizes (see Table 4.1). We recruited individuals involved in building AI components for user-facing products; mainly, UX professionals and AI engineers, data and research scientists, and managers. Starting with university alumni and other industry connections, we used snowball sampling to recruit participants through referrals to other professionals. Before the interview, participants completed a consent form, and in many cases, also sought approval from their

Organization	Interviewee(s)	Business Model	Size of Organization
O1	S1	B2C	1,000 – 5,000
O2	S2, S4, M3	B2C	10,000 – 50,000
O3	M2, R2, U1, U5	B2C, B2B	$>$ 100,000
O4	M1	B2B	$<$ 100
O5	D1	B2B	$>$ 100,000
O6	S5, R1	B2C	$>$ 100,000
O7	U2	B2B	$<$ 100
O8	U6, D2	B2B	$>$ 100,000
O9	S3	B2B	$<$ 100
O10	D3	B2C	1,000 – 5,000
O11	U3	B2C	10,000 – 50,000
O12	U4	B2B	100 – 500
O13	S6	B2B	5,000 – 10,000
O14	S7	B2C	$<$ 100

Table 3.1: Each organization is listed with interviewees by role (S = Software Engineer, U = UX Professional, M = Manager, D = Data Scientist, R = Research Scientist) and a brief description.

company's legal team. We conducted all interviews through video-conferencing, with each interview lasting about 60-minutes. We audio-recorded all interviews, and a study coordinator took notes.

3.2.2 Analysis

We used a third-party service to transcribe all interviews and conducted qualitative coding for analysis. We determined that deductive coding based on the affinity groups may be biased and fail to reflect participant responses. So, we led inductive *in-vivo* analysis beginning with an initial review of the interview notes. First, two coders independently coded five transcripts and then worked together to develop an initial codebook. This revealed vital themes, such as the use of prototypes, multiple workflows, and friction in collaboration. The coders then analyzed the remaining transcripts using the same codebook over two passes [63]. A memoing activity followed this to synthesize the findings across transcripts [29]. The analysis focused on how collaborative teams develop human-AI applications.

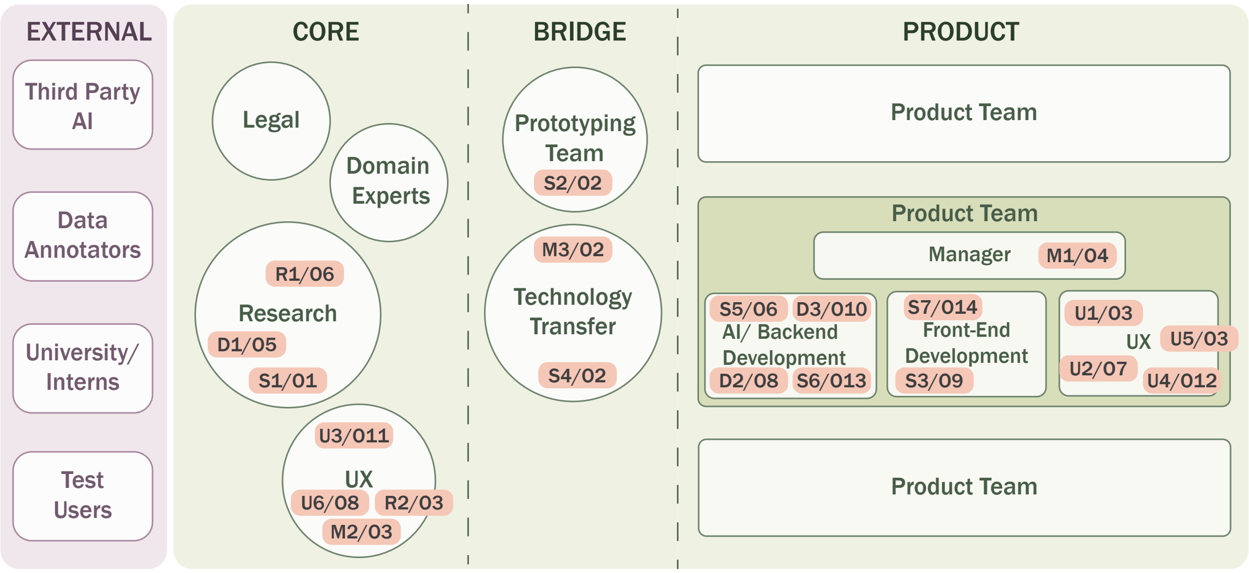


Figure 3.3: Generalized Organizational Structure of Teams in Human-AI Application Design and Development. Interview participants are overlaid onto corresponding teams (S = Software Engineer, U = UX Professional, M = Manager, D = Data Scientist, R = Research Scientist). ”O” denotes the organization number.

3.3 Findings

3.3.1 Design-Engineering Boundaries Hinder the Cross-Cutting Needs of HAI

Across all organizations discussed in the interviews, a separation occurs between individuals who *conceptualize* AI capabilities and those who integrate those AI capabilities within end-user products. As shown in Figure 3.3, many large organizations have dedicated AI research teams (primarily computer scientists) who explore novel AI capabilities and techniques. However, the technology itself may be only partially motivated by real end-user needs. For instance, a research scientist may investigate a new facial recognition algorithm for predicting people’s age and gender. Product teams are not typically involved in this initial AI exploration process. However, once the technology vision is achieved, research teams join in with different product teams to identify product use cases for applying AI innovations. Large organizations may also have intermediate technology transfer teams that envision product and human uses for AI innovations to support a research-to-product pipeline. Rather than develop core AI capabilities in-house, smaller organizations may rely on third-party AI providers (e.g., Amazon AI Web Services) to add new AI capabilities into product features. Outside of core research and product teams, the AI development creation process commonly requires support from domain experts and data annotators. These teams tend to be external to the organization. Further, product teams may consult with legal representatives about ethical data collection and data privacy issues. Organizations may also have a pool of beta-testers available to evaluate new features during development.

The interviews revealed how team structures and separation of concerns (boundaries) between differing roles and expertise *hinder* human-centered AI design. Specifically, team boundaries introduce three central challenges to HAI design, including (1) knowledge blindness in AIX design, (2) premature specification of AI and UX, and (3) boundary friction in aligning AI and UX.

3.3.1.1 Boundaries Introduce Knowledge Blindness in AIX Design

HAI guidelines recommend that AI capabilities should be motivated by human needs and should align with human behavior, task workflows, and cognitive processes. However, the boundaries between core AI developers and UX designers limit possibilities for creating human-centered AI from the ground up. Given the novelty of AI, researchers and engineers are motivated (and incentivized) to explore AI capabilities independently and without regard to products and end-user needs. As M3 describes: *"... research coming up with new cutting edge state-of-the-art techniques for doing something that the product team wasn't even thinking about, or users aren't asking for, because they hadn't thought that way."* This boundary separates developers from end-user product teams and introduces end-user blindness about product users' needs and concerns.

On the other hand, product teams – specifically UX designers who advocate for end-users in design specifications – may lack an understanding of AI technology capabilities. As a result, UX designers appear to either lack trust in or over-rely on the AI technology, which becomes manifest in the UX design for AI. As R2 puts it, designers tend not to automate things that they could be automating: *"There's under trusting where it's like oh actually you should let the algorithm make a suggestion, maybe offer a choice, maybe you should trust it more than you do."* R2 further adds that in other cases, there is over trust on what AI can and cannot do: *"... then other times, especially when you get into the cases around anthropomorphism and things like that, people really overshoot their trust and think yeah this is going to be great no matter what happens, and I don't need to worry about it."* A consequence of the black-box approach to design is that designers themselves lack clarity about the AI output. This makes it challenging to design user experiences that align with end-user mental models and expectations. In referring to AI as a material for UX design, M2 comments on designers needing rather understand the capabilities and limitations of AI after it is conceptualized:

> "It used to be that UX designers just made static mocks and there was screen-to-screen flow. But now designers need to understand probabilities. The screen to screen flow has always been the happy path, right? but the probability that the end user makes it

> through that path is lower. Just the probability curve on any journey is now much more complicated and is much more branched and probabilistic, just because its driven by a sequence of machine learning models, and the probability space is much more widely open. You need to understand the failure cases, understand confidence and how you deal with confidence scores, how you threshold. You need to be able to understand the grain of the technology. Just like you understand wood and its material structure, you need to be able to play with machine-learning model in order to understand the edges and the possibilities." - [P11]

3.3.1.2 Traditional Software Practices at Boundaries Impose Premature Specification of AI and UX Designs

In conventional software design, UX professionals work with end-users to define the requirements for the application. However, because of knowledge blindness, on their own, designers and engineers may not be able to fully define specifications for either the AI model or the UX interface. Yet, our interviews identified the tendency to define AI and UX specifications independently because of the work-role boundaries and lack of coordination practices. This problem takes control away from designers attempting to craft the end-user's experience. In highlighting this concern, S5 reported that *"... we kind of came up to them and said, 'Hey, we want to do this thing that's going to be powered by machine learning and you don't have control over what that's going to be, because it's going to be powered by some kind of algorithm... You guys onboard?' We got a lot of hesitation at first. But it was very much pushing everything into this direction of, a place where they can't have fine grain control over every single detail. And to some degree or another, leaving it up to some kind of algorithm to pick what content you want to show and where... "*

Across many interviews, designers expressed frustration in trying to design the UX around an independently created AI specification. For instance, in one of the sessions, the AI team developed a nuanced tagging schema for media content and handed it off to the UX designer to integrate into a voice assistant. The designer (U1) comments on the challenge in integrating UX around predetermined AI specifications, noting the extensive rework and feedback required to retrain the AI tagging model in a way that meets their understanding of end-user needs:

> "In the first meeting, they [AI team] were just saying 'We need to add this [categorization] on the screen, can you find where is the right place?' Then I need to work a little bit backward to say there are constraints on wordings, and voice UI had another layer of constraints, so I need to understand how this information fits into the screen we already have. They have their own taxonomy on what kind of information they are kind of looking for, but for users, it doesn't evoke a response ... The [label] semantics

> is not on the same level to what we already have.”- U1

Similarly, AI engineers also find it challenging to implement desired AI features when the UX is already specified in great detail without AI involvement. As previously noted, AI models (unlike conventional applications) are challenging to build to specifications because their behavior is dynamic. This makes it difficult for engineers to create AI technical specifications from design requirements alone. S7, an AI engineer, comments about their frustration in the coordination and hand-off process between UX design and engineering:

> ”. . . they would hand that [design document] off to the engineer and say ‘Implement this.’ And of course my reaction to this was ‘This is garbage.’ This does not reflect the appropriate architecture for implementing this thing. It felt particularly extraneous when it got very granular, and it was not the best medium for describing the desired behavior. Because the designers were not technical really. This is not a good reflection of how the actual software engineering is going to happen. And I was like, ‘Stop trying to do my job for me.’ ”

The problem of AI blindness among designers arises from the role boundary created by professional expertise. By advancing UX design independently from AI teams, UX features become ”set adrift” from the other source of constraints for end user’s needs : the AI model.

3.3.1.3 Boundaries Restrict Access for AI and UX Collaboration

Because of differences in the design and engineering processes, there is no clear understanding of how human-centered design requires alignment across both tasks. For instance, U6 expressed concerns that the UX team was not involved in the training data annotation process – the core of how end-users experience the AI. According to U6: *”it seemed very odd to me that as designers we were not invited to the annotation session. So, we had to invite ourselves to just talk to domain experts. . . ”* U6 further commented that *”. . . for engineers, their approach is more like ‘the machine is going to figure it out.’ We could be talking about health or elephants in the circus, and it is all the same to them . . . ”*

Across the interviews, other collaboration challenges surfaced. First, the core responsibilities for UX professionals are defined differently from basic AI research. In addition, the time needed to conduct user research is viewed as out of sync with AI research progress. For instance, U4 comments that *”we don’t necessarily participate as much in that whole AI thing, but the thing is because we’re also trying to make sure that we’re doing user research and participating in that”*. S4, a research engineer in a different organization, offers their perspective on collaboration: *”. . . they [UX] might complain after the fact that they weren’t early enough, but on the flip side if we* *try to involve early then they'll say they're busy doing x, y, and z. In my experience, it's not always practical"*. Acknowledging the added time it takes to conduct user research, M3 comments:

> ”You obviously need a human need, or you're not going to have anything worthwhile, but the reality is in most of these companies there are in-flight research efforts that are happening on basic capabilities, and it's not like you can say, 'Okay, everybody stop what you're doing until I find you a human need, and then you may start working again.' It's just kind of absurdity." -[M3]

In smaller organizations that work with AI as services, boundaries severely challenge the design of AI behavior and presentation for end-users. In working with third-party AI, M1 describes that designers often have to engage in the laborious activity of translating AI output into user-friendly labels: *”. . . the label that the database has for the data may not be the same as what your end-user understands it to be. So, understanding there's a difference between how an engineer labeled it in the database, versus how you might want to show it on your UI to the end user. . . we would look at the raw JSON files and create our own labels . . . ”*. The lack of collaboration forced UX designers to intervene in the AI model specifications to prevent later HAI issues.

3.3.2 Data Needs Change How HAI is Designed

In conventional applications, UX designers analyze and synthesize higher order requirements from end-user data. However, in machine learning applications, data points (i.e., examples) *are* the requirements. Therefore, the process for designing HAI is necessarily different from conventional UX design. The interviews identified a range of stakeholder interactions and data constraints for generating AI requirements.

3.3.2.1 User-Data Informs AI Requirements, then User Needs

Because of the technology-first approach to HAI design, AI requirements appear to drive end-user needs identification via data annotations and validation tasks. As reported by three participants, this workflow aims to optimize the AI development process. When exploring new AI capabilities, researchers don't always understand what types of data might be needed. Requirements about data and its characteristics, such as variables, data types, labels, and number of data points, evolve through a ”trial and error" approach. That is, researchers start with an initial, small dataset to train the model. For early stage data collection, organizations may have internal data collection and logging apps (e.g., one that collects gesture data while using the phone) that can be deployed across different teams. This lowers the cost for data access. There is often an "unwritten agreement [S5]" that development teams will provide data for AI development purposes. Based on model behavior and outcomes, UX researchers may redefine data needs or collect additional data to test for AI robustness. In this process, AI researchers prefer quickly collecting data as needed to train their models. As S5 comments: “you have to spin up a dedicated user study program, go through a lot of process, a lot of review, it’s a whole lot of bureaucracy to get that kind of rich data collection”. Therefore, AI researchers work with pre-existing data sets or ‘minimum-viable-data’ collected from within their UX team and then gradually increase scope of data over time:

“For collecting data, we will start from people within our team and do a first pilot testing [of the AI]. If it works well we will increase the size of data. For example, we will recruit people from a different project team so they are not biased to our data needs. And if it continues to work well but we still need more data, we will start looking for external partnerships to collect data from our target users.”[S1]

Once the AI capability and data specifications are determined, UX teams may work with end-users and customers to annotate and label user data for the application design. As M2 comments: *“if you want a certain data structure with a hundred hypothetical labels, you can show that to users and get sentiment on that…”* Further UX designers commented that such a partnership requires careful consideration about privacy and content ownership, as well as communication about benefits to customers. In planning the labeling task, M3 comments, “gather data without causing a ruckus, invading their privacy, or taking creative material in a way they would object.” Further, AI engineers also emphasized the need for clear communication about how customers (who are assisting with data labels) might benefit from their contributed data. Because of the way user inputs are elicited, S6 commented on end-users hesitant to provide information for labeling tasks:

> “We asked customers [to label the data], but it wasn’t good enough for our use. Anecdotally, I think the people who are being asked to label weren’t sure how this information is going to be used. I think there was some hesitation because it wasn’t tied to their day to day metrics or goals. I don’t know if there was an element of fear of automation…” - [S6].

As a consequence of AI model needs, data collection from end-users appears to occur more incrementally and less formally than with conventional applications.

3.3.2.2 Data Tools for End-User Needs Elicitation

Given the significance (and multiple roles) of data in HAI design, data collection and data annotation tools are essential for gathering end-user requirements. Consequently, engineers develop custom tools for collecting needed data. For instance in a photography application, M2 comments on creating a tool for eliciting end-user needs around image quality. According to M2: *“…figuring* *out what makes a quality photograph for a specific user is a challenging problem. There's no model for it, so that team came up with this method where they collected... Let's just say they left the shutter open for an hour. Then, the team actually created an annotation tool to be used by a small set of professional photographers, very high-quality rater, kind of, scenarios. They would go through, and they would select what they thought were the highest-interest sections of that video stream."*

Often, these tools are designed based on data needs and labeling, and are optimized to lower the engineering cost for data cleaning and transformation. According to S1: *"A lot of times, our problem is not generalizable, so we build our own tools in house."* Such tools are designed with debugging as a primary objective. For instance, the data collection tool may explicitly ask participants to start a session and perform some task, or prompt participants to validate whether or not the right label was detected (e.g., labeling sensor-based activity detection). In this workflow, engineers also reported striving for ''clean'' data by removing outliers and noise to improve model performance. This may lead to a idealistic versions of data for AI model exploration that omits features potentially relevant to requirements and omits real world use.

UX designers in the interviews acknowledged that labeling can be tedious work, and expressed empathy for people charged with labeling the data (e.g., *"there are overseas sweatshops where people are just filling in Mechanical Turk surveys day in and day out, figuring our whether the image has a dog. As a designer with all the empathy in the world you have, you feel really bad for those people"*[S2]). In one interview, the designer reported visiting those performing labelling on-site in order to understand their pain points, and to run user studies with them to evaluate data annotation tools.

> "We wanted annotators to create object segmentation boundaries on images by drawing polygons. To design the tool, I visited [location] and asked the annotators to generate labels. From these trial runs we noticed that using the keyboard was essential for a good UX, and they needed ways to undo or edit polygons. Based on this we did a focus group to know how we can improve the labeling tool."- [S3]

This example illustrates change in the nature of data, how it is used, and how it is collected for the design of HAI systems. UX designers are learning to provide new forms of data in new process timelines as driven by AI model development.

3.3.2.3 Authentic Data for AIX Evaluation

As with technical evaluation of AI models using a"holdout" dataset, in many instances, usability testing of HAI applications requires that end-users supply their own data based on their personal experience history in a domain. This allows end-users to provide feedback about AI behavior from their viewpoint as experienced within their own situated contexts. As R2 puts it: "The best mock for AI is a lot of times human. We really try to use people's own content. This is the thing, if I look at photos of my friends and family, I'm going to have an emotional reaction, I'm going to have an authentic experience there." Consequently, AI model design requires continued evaluation and feedback from diverse end users with personal experiences in a task domain. However, within existing design and development workflows, constant engagement with end-users (ranging from novice to domain expert) is challenging for UX designers to accommodate. In describing this challenge, S5 comments:*"User studies, especially things of this nature, like, getting around a lot of our privacy constraints tend to be difficult, which that's a whole another like, can of worms you probably don't need to attack right now."* S5 points out that evaluation, especially for recommendation systems, require access to user data and requires time-consuming review for privacy compliance.

In addition, teams find it challenging to come up with the right metrics to gather feedback on AI experience design. According to D3 *"To me, evaluation is still very, very hard. And especially I think maybe more subjective evaluation too in terms of the quality or how enjoyable was the experience?... if you were using the measure of how many items you interacted with or how long you engaged, it would feel like the one that was a five-item engagement was more successful than the two-item engagement, where actually they [end-user] didn't really think that at all." (D3).* A lack of well-tested metrics makes it hard to run deployment studies to gauge end-user expectations and trust. These challenges are amplified in evaluating AI behavior over time, especially for learnability through end-user feedback.

3.3.3 Collaborative Design Processes with Constant Co-Evaluation

In response to the expertise boundary and data role challenges, participants reflected on how they reduced friction to facilitate engagement across teams. These workarounds involved a variety of boundary negotiation artifacts to support communication and knowledge sharing, collaborative prototyping and design negotiation, and design evaluation and feedback.

3.3.3.1 Bridging Knowledge Boundaries between Designers and Engineers

Communicating about end-user needs with engineers: In conventional software workflows, UX designers rarely share raw end-user data and low-fidelity representations with engineers. As S2 puts it, "they are blindsided to sketches, wireframes or any other low fidelity prototyping. They only understand high-fidelity prototypes, even more so if it is interactive." However, the interviews revealed that sharing low-fidelity representations is effective in centering the end-user within AI model design. For instance, UX designers reported sharing raw user-data and co-creating personas with engineers in order to help them think about training data needs. This requires a larger data collection program and generates needs for different data collection tools, types of end-users to recruit, storing and processing data, and collecting data preserving privacy and ethical concerns. In describing their approach to ensuring the representativeness of differing end-user groups in collected data, U5 comments:

> "Often look out into the world first to see what information there is about existing groups, and then evaluate for myself, do these groups make sense or do I need to make different groups. I have done all of the user research and come up with groupings on my own, and then brought them back to the team. Then I talk it through with the PM and the engineers what the value of different user segments are, why would we want to prioritize the different users, why are they important to the company. It's always nice to pair that kind of survey work to understand the broader population, with those interviews, to kind of figure out the best way to group the users based on the goals of your team"- [U5]

With HAI, the task of anticipating relevant differences in end-user populations impacts not only the UX design, but also the behavior of the resulting AI model through training.

Another change in UX designers' work for HAI occurs when designing interaction or task workflows. S7 reported that sharing storyboards offered flexibility and control in mapping user needs to AI features and implementation logic.

> "I find storyboards very helpful. Storyboards or other documents that get into describing what the purpose of the behavior is, what the desired user experience is without getting into the engineering. I think of it as a sort of comic book illustration of what the user experience should be and what the system's reaction should be in different interactive situations. It was like sort of the key expected traversal through an interaction, and then maybe some of the most likely other paths about what experience you want the user, and the [system] to have together. Here is a situation, and what should happen over the course of this interaction. And I don't mean to seem territorial about this, but it's really useful to have back and forth with the people who are trained to think carefully about user experience." - [S7]

UX designers reported innovations in their methods for eliciting end-users' authentic responses to potential AI capabilities without engagement with the model itself.

Communicating AI Capabilities to Designers: Participants reported varied strategies for sharing AI capabilities and details about implementation (such as assumptions and logic) with UX designers and domain experts on projects. The intent is to resolve technology blindness and to facilitate collaborative design and feedback. As S6 comments: *"If we don't adequately communicate to* *designers, they fill in the gaps with their own theories and its not clear what input needs to be provided in order to get the desired results.”*. In one scenario, AI researchers reported working with university interns to develop a conceptual prototype of an AI feature. Here the goal is to (1) demonstrate a new capability of AI within an application context, and (2) define a design space for UX researchers to think about the experience. As S5 describes: *“We got something tangible enough that we could actually go talk to a designer and be like ‘Here’s what we’re thinking’. . . we started letting them play around with it, and said, ‘Try it out for a week and tell us is this better than the old way that we’ve done things.’. . . it also broke the problem down such that the designers understand, here’s the benefits of where the machine learning can be applied.”* Once UX designers understand the AI design space, they are able to collaborate with researchers to explore end-user needs using the prototype as a design probe.

In other cases, AI researchers may identify a new technical capability but find it hard to define its context of use. In such cases, UX researchers need to first understand the technology and then identify its benefits for potential end-user experiences using prototyping approaches (as suggested earlier). As [M2] describes: *“A lot of times, people are, just kind of, down in the weeds, really deep and get a little lost in the day-to-day work. UX teams can actually bring a little hope to those folks and give people a target, and really paint a picture of that through design visualization, whether that’s making a movie or just making a series of mocks, or building an experiential prototype, or something like that, really help land the tangibility of something that’s pretty deep and complex. Sometimes, it’s the light at the end of the tunnel. . . ”* However, this is challenging process requiring the UX researcher to possess some technical background, as R2 humorously comments: *“I joke that half my job these days is just being an API layer in between UX and research science”*.

Further, two participants, both project managers, emphasized the value of UX friendly machine learning tools for creating experience prototypes. Specifically, these tools allow designers to take ”off the shelf” ML models and work with real end-user data to demonstrate an envisioned AI feature. Using actual ML tools also mitigates the danger of setting or communicating unrealistic expectations with AI mock-ups. According to R2: *“I cannot believe I am saying this, but years ago, it was all very vision-y. It was all mock-ups and animated videos and frankly, there were very different opinions on what the technology could do. Those that are more optimistic won out, but then they were proven to be too optimistic in a lot of ways. Surprise, surprise! So now we are actually getting to a place where we can do a bit more realistic prototyping because we have these AI prototyping platforms like Runway ML. You have UX engineers who can get some of it actually working. It is ok that it isn’t our specific computer vision model, but you can plug in one of the off-the-shelf models and show that this is what it would feel like when it recognized a flower or whatever. . . ”* The existence of ML tools applicable in novel applications provides AI prototypes with some functional capability, allowing more AI feature exploration without the expense of building AI models that are eventually abandoned.

3.3.3.2 Collaborative Design of HAI Prototypes

By identifying ways to bridge expertise boundaries, designers and engineers reported working towards collaborative creation of prototypes including data and labels, AI Model behavior, implementation, and end-user experiences. For instance, HAI guidelines recommend defining data labels and annotations by consulting with expert users. In the interviews, UX designers and engineers identified multiple ways to work with domain experts to co-design labels. When available, participants reported continuous engagement with in-house domain experts throughout the data design process. For instance, the data scientists generate the necessary queries for exposing different types of data requiring labeling, engineers define ML constraints for labels, and UX designers and domain experts generate and validate labeling schemas (i.e., rules for assigning labels to raw data). As D3 describes: *"So labeling, it was a collaboration between the four of us. I was the data scientist who looked at a lot of the data. There was a machine learning engineer who had worked on voice assistants and had a lot of experience. And we also worked with a data curator for labeling…that is how we came up with an initial labeling scheme. The data curators are domain experts who work a lot on labeling data for personalization models. The curators would do a lot of quality judgment work too…"*. A second collaborative process identified occurred when data scientists find pre-existing datasets they re-purpose for their AI needs. In this workflow, data scientists work with domain experts to clean data, identify variables for prediction, interpret data analysis results, and perform labeling. As D2 describes it:"we would be talking to meteorologists about how to adjust variables, and create flag variables, so if it is above this temperature or dew point, we would categorize it…". This collaborative process happens through sharing CSV files, python scripts, and visualizations.

Further, creating experience prototypes combining AI capabilities and UX needs requires close collaboration between designers and engineers. In the case of Wizard-of-Oz prototypes, UX designers gather end-user data and work with engineers to generate outputs as well as understand the logic behind them. This is essential to understand the unanswered questions from an engineering standpoint, plan the type of user study needed, and design the presented experience of the prototype for end-users. As U3 describes: *"Let us say I am doing food recommendations. And I want to tell users why something was recommended. It may be because they are liking a few restaurants, or they added items to their shopping cart, or maybe it is because of past orders. It is a Wizard-of-Oz prototype where I first get users' data. Then I get the model output from the data scientist and work with them to understand the model labels and explanations. The data scientist wrote down all the equations and explained it to me very clearly. They showed me how the weights were set, and we discussed things we need to know from users, whether to do an A/B testing or a walkthrough…"*

Engineers also support UI designers through annotations on wireframes about what is happening behind the scene. According to S6: *"I added annotations on the side about what is happening behind the scenes like an API is being called. Then as an example, I would [annotate] for the API what output it comes back with... I use Balsamiq [UI prototyping tool [17]] because I think it lowers the barrier of what can be a design tool and you don't need specialized knowledge to communicate that idea."* These efforts by developers indicate efforts to support greater collaboration and extension of expertise across boundaries.

3.3.3.3 Design Iteration with Constant Evaluation

The interviews revealed that in successful HAI development workflows, evaluation happens frequently using incomplete prototypes still under development. In fact, participants reported that this form of evaluation is necessary when the user experience is co-evolving with AI development: " *I think the process that works best is fairly tight review cycles with the actual evolving behavioral artifact."*(S7). In such cases, functional prototypes allow testing and gathering feedback about AI behavior. U5, a UX researcher who helped set up a prototype testing program, describes the process as *"I think tie your fidelity to testing as you're getting closer to what that end product might look like. You might have to use it on hacked together hardware or something a little bit different, like it might not be as smooth as the end product will be, but as you're getting closer and closer to that real product experience you're able to just kind of dig more into the nuances of the products, but then also just kind of unearth some of those additional considerations that you might not get from just talking through."* One challenge with this process is communicating with designers about what is implemented and what is not, and what type of feedback they need to provide. According to S7:

> "I mean, you just have to make them understand. I think that is part of being in a non technical role is understanding enough about development. So you need to tell them 'listen, what we are showing you today is two weeks of work. Here are the things that it doesn't have but it will have. We don't need feedback on the fact that it doesn't have sound effects or graphics. What we need feedback on is , is this the basic kind of interaction you want? Does this look like something that is going to solve the problem? Trust us. We will get back to polishing it, that not what we are looking at at this stage.... The other side of this is that as an engineer you have to be able to interpret and filter the feedback that you get. Because it is inevitable that people are going to be giving you small, more fine grained feedback and what you really want is big directional feedback. So you capture that, file it away for later." - [S7]

Identifying key functions to test, and why, along with which functions are missing and why they don't matter at this moment, requires UX designers to learn a great deal about AI technology in order to support its data needs.

Early stage evaluation of model behavior: In the early stages of development, engineers may make certain assumptions about AI behavior. Frequent evaluation allows UX researchers to provide early feedback about these assumptions. As S7 describes: *"... as I was implementing this feature and I ran into this problem of how to handle this use case? ... Here is the guess that I made, but let us talk about whether that was the right choice. As things were getting built, we would look at the running prototypes and be like, 'Do we like how this plays? What is missing?'... "* Here, S7 describes how this approach is more suitable for AI development compared to having a black box prototype provided by the UX designer. Similarly, for AI perceptual (e.g., computer vision) interactions, UX designers may provide an initial set of desired interaction gestures. Then, during development, designers and engineers evaluate the feasibility of those interaction gestures using prototypes and discuss alternatives. S1 explains that *" The designer will say 'we want ten different facial expressions for this model'... we start from there to build the backbone of the interaction, and then we iterate through it... we call like grayboxing... there are three facial expressions where it's just really hard to get that right, it is not going to perform very well. The other said seven is fine. So, in the process of testing we find out, there are two other facial expressions that are not in the original ten expressions that can perform pretty well. And so we will tell the designer that these three we will need to cut it. But if you want there are two more gestures you can add into your interaction... "*

In a different scenario, evaluation with prototypes helped engineers determine the optimum algorithm for a problem (user need) they are trying to address. In describing the iterative process of model comparison to find the best approach, S2 explains that: *"... the process involved 20 different prototypes I had to build for all the different algorithms we've tested on."* Further they describe that the prototypes expose the actual logic using visualizations for test users to evaluate: *"I did the visualization of the user uploading an image and the palette created so that we know how the algorithm is working under the hood. Because AI is a black box, we need to have some transparency in there for the user to understand. You show the palette. Once you have the palette, it will do the search and return the results. For each of the result, I also show the pre-indexed pallets which we use to compare with others once you have that exact side by side you can do like simultaneously see."* This allowed the UX designer to do a comparative evaluation *iteratively*: *"every new week when we have a new algorithm, we compared to the existing best and see which one is still the winner and then that will compete with the next algorithm. And so we find which algorithm is the best every time. That is how we reach to find the one which we shipped to productions."* In other cases, engineers may expose a set of knobs on functional prototypes for UX researchers and product teams to tinker with and figure out the right parameters for the model: *"in your initial* *prototypes you have that dropdown kind of settings panes where you expose all the knobs and let the product managers tinker till they finalize the threshold that works for them"*.

As an alternative, the data scientist may provide domain experts with a spreadsheet containing rules and assumptions made in building the AI model. The expert then annotates changes to the rules for updates of the model. According to D1: *"There are rules and codes we have that we use for making recommendations. We would list out the rules so the domain experts could look at it. We started to give them more accessible tools like sharing a spreadsheet where they could do some input and we could take that in to ingest into the system. They had the ability to flag, add notes and annotations [about model output]."* This process allows domain experts to participate in specifying AI behavior at a conceptual level, providing direct input into AI specifications.

Evaluating interpretability features: Designers mentioned taking an iterative prototyping approach to determine the right level of abstraction for AI output. Working with front-end engineers, designers create and test functional prototypes with different output formats. In discussing their process for showing output probability to end-users, U2 comments that *"There are two versions we iterated. The first one is to show the possibility as numbers. If I have ten patients and nine of them have 100 percent, and only one shows 20 percent, it might confuse a user because a number is really hard for a [end-users] to understand…The second version we actually tried was high, medium, low possibility. So that turns out to be more positive by the user."* Here U2 mentioned working with domain experts to translate percentages into categorical bins, such as "high", "medium", and "low".

Evaluating data and model for privacy and ethical concerns: To evaluate privacy and ethics during data collection, AI engineers often collaborate with members from the legal team. Many interviewees described this as a collaborative process in which engineers do a walkthrough about what data is being collected and why. Then, they engage in discussions about alternate data sources in case of privacy violations, and how to collect data in a privacy-preserving way. As described by S5: *"all the data collection has to go through a privacy review … you sit down with one of them, you walk them through, here is the data we want to collect, here is why we want to collect it. They do some discussion about, is all this data necessary, can we do different ways to interpret it?"* This process often involves sharing compliance documents and details about protocol and data, and a legal team may draft a privacy statement for end-users to review.

Evaluation in the wild: When a fully functional prototype is available, UX researchers may conduct deployment studies with test users to evaluate how the model performs in the real world. M2 describes this process as: *"Anybody can basically download [the] app and try it out, That's how we collect data a lot… its very easy for a UX researcher to go back and say, 'We're seeing this fail for this use case,' or 'for this population,' and just go back to the team and it's an open conversation about the limitations of the current model and how to adapt…."* Further, UX researchers may conduct a longitudinal evaluation with functional prototypes. According to U5: *"... doing longitudinal research is really helpful ... if it is something that takes a bit of ramp-up time, giving the people you are testing with time to spend with it, to see where it lands and how useful it is over that time."* Further, in communicating to users about longitudinal testing, U5 comments that *"I think some of it is just product transparency, it would make sense for me to just be like, "Right now we don't know anything about you, but come back as you use this app over the next couple of weeks. We will start to produce better recommendations for you." So keep checking back, because otherwise, I think you might make assumptions that it is never going to work or things like that. So I think transparency can be really helpful in those situations."*

3.4 Discussion

Human-centered AI is a multidisciplinary endeavor. Previous HAI guidelines span forming expectation models for human users about AI, user interface and interaction design, model design and implementation, and designing training data needs. However, as our findings show, work-team boundaries introduce numerous challenges to knowledge sharing and collaboration in HAI. Current HAI workflows are primarily "AI-first" in that the AI capabilities are developed first, and then UX designers are brought in to create the application experience around the AI. In an "AI-first" process, AI engineers envision, conceptualize, and develop AI in the absence of end-user influences.

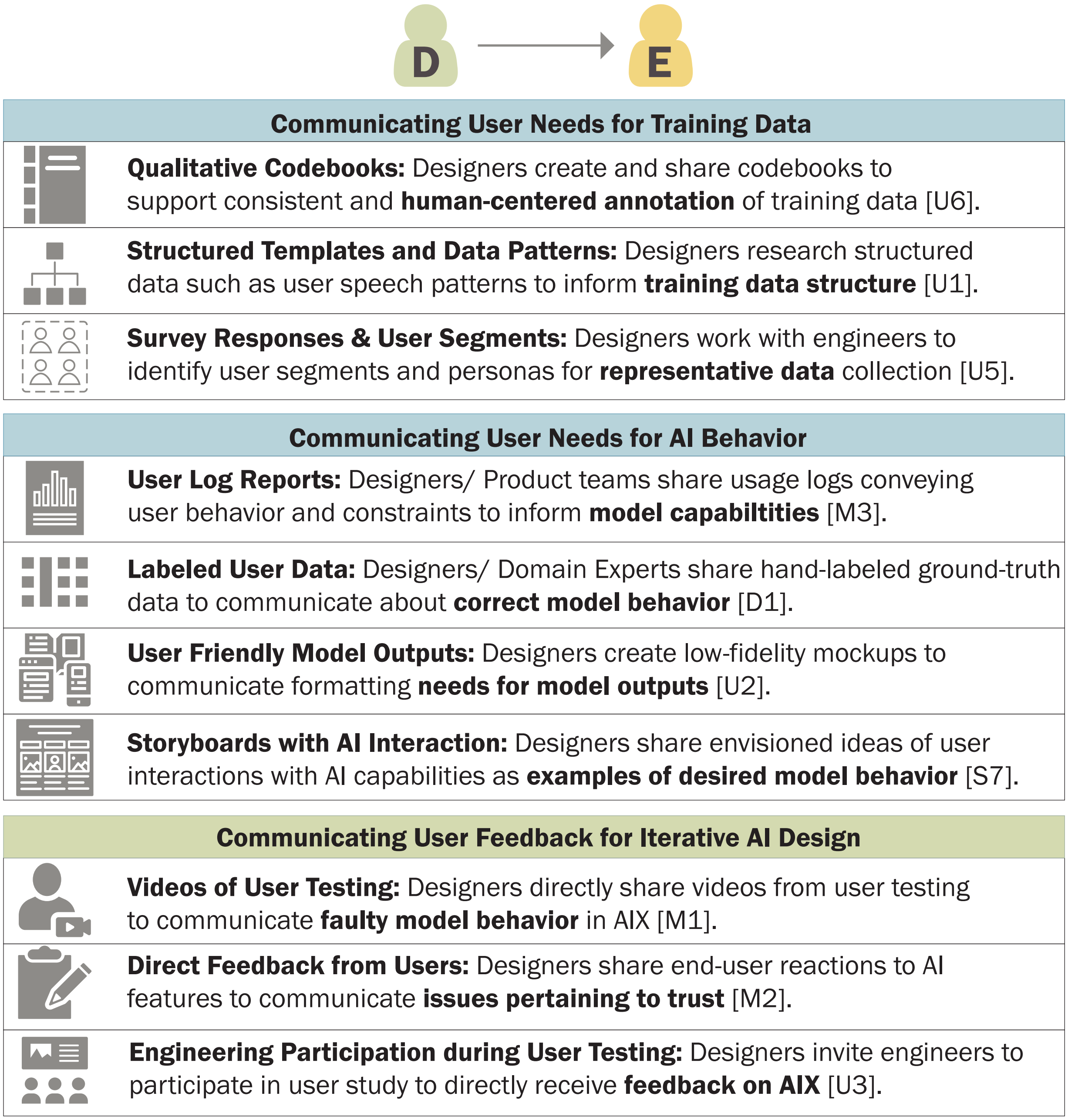
Consequently, AI specifications are determined with minimal inputs from potential end-users, resulting in less than ideal AI experiences. A contributing factor in this "AI-first" approach to HAI design is that UX professionals lack familiarity with AI technology as well as the means to design AI experiences for human needs. Technology blindness for designers results in their limited participation in the AI development process and leads to premature AIX specifications that are challenging for designers to implement in practice. For instance, the designer might later specify "exact" AI behavior and interactions without the ability to account for AI uncertainties and failures. The probabilistic nature of AI makes it challenging to deliver designed experiences consistently. As a result, engineers cannot build AI components to exact design specifications. However, in contradiction to established software development principles of information hiding and modular design, our interviews also revealed workarounds and collaboration practices for designers and engineers. Here, we discuss two critical findings to support collaborative HAI design: (1) leaky abstractions and (2) delayed specifications. Based on these to findings, we argue for the need for designers and engineers to co-design AIX. These principles answer our research question on how teams can operationalize HAI guidelines in practice.

3.4.1 Leaky Abstractions are Necessary for AIX Design

To accommodate differing expertise and division of tasks in software development workflows, teams typically adopt the practice of information hiding or separation of concerns [189]. Concretely, software is divided into modules consisting of an outward-facing *interface* and internal *implementation* [187]. Teams agree on the interface or API specifications about what functions the module should expose, what inputs it should require from the caller, and what outputs in which format it should return to the caller. The module's interface offers the necessary *abstraction* to callers without exposing unimportant (and potentially complex) implementation details. In fact, any "information leak" about implementation is considered a 'red flag' in software design [187, 110]. However, as our findings show, strict abstractions introduce knowledge blindness for both designers and engineers. Designers and engineering teams find it challenging to define or agree on the interface without deeper knowledge of the implementation. The separation of concerns that allows efficiency in collaborative software development disregards the complex dependencies between abstraction and implementation [2]. Instead, our findings show that *leaky abstractions*—those instances in which teams disregarded software abstractions and exposed low-level design and implementation details were critical in bridging knowledge boundaries and supporting the collaborative design of AIX.

As reported in our findings, to shape AIX around end-user needs, designers share low-level design details with AI engineers (see Figure 3.4). For instance, to inform training needs, designers provide details about personas that emerged from surveys, shared qualitative code-books with terminology, definitions, and guidelines for training-data annotation, and raw end-user data gathered through UX research processes to inform data characteristics, representativeness, formatting needs for AI's training data. Further, designers provide 'examples' of desired AIX interactions through storyboards, prototype interfaces for task workflows, spreadsheets with ground truth data, and even interaction logs from existing non-AI software use. These artifacts communicate to engineers about needed AI behavior. Third, given the challenges in articulating and reporting feedback about AI from end-users, designers share raw feedback from user testing through videos and direct observational notes, and invite engineers to participate in end-user evaluation sessions. These new collaborative practices characterize the nature of "information leaks" about end-users and design to inform AI development. Designers offered technical representations such as qualitative code-books and epistemic design objects (including storyboards and prototypes) as shared representations for AI and UX specifications. Through these representations, designers renegotiate the design-engineering boundaries and give inputs about model behavior and training data. These design artifacts help engineers situate AI decisions within the broader context of AIX design.

Similarly, engineers reported numerous instances of leaky abstractions and new collaboration practices to surface AI implementation details for designers. As shown in Figure 3.5, abstrac-



Communicating User Needs for Training Data

Qualitative Codebooks: Designers create and share codebooks to support consistent and **human-centered annotation** of training data [U6].

Structured Templates and Data Patterns: Designers research structured data such as user speech patterns to inform **training data structure** [U1].

Survey Responses & User Segments: Designers work with engineers to identify user segments and personas for **representative data** collection [U5].

Communicating User Needs for AI Behavior

User Log Reports: Designers/ Product teams share usage logs conveying user behavior and constraints to inform **model capabiltities** [M3].

Labeled User Data: Designers/ Domain Experts share hand-labeled ground-truth data to communicate about **correct model behavior** [D1].

User Friendly Model Outputs: Designers create low-fidelity mockups to communicate formatting **needs for model outputs** [U2].

Storyboards with AI Interaction: Designers share envisioned ideas of user interactions with AI capabilities as **examples of desired model behavior** [S7].

Communicating User Feedback for Iterative AI Design

Videos of User Testing: Designers directly share videos from user testing to communicate **faulty model behavior** in AIX [M1].

Direct Feedback from Users: Designers share end-user reactions to AI features to communicate **issues pertaining to trust** [M2].

Engineering Participation during User Testing: Designers invite engineers to participate in user study to directly receive **feedback on AIX** [U3].

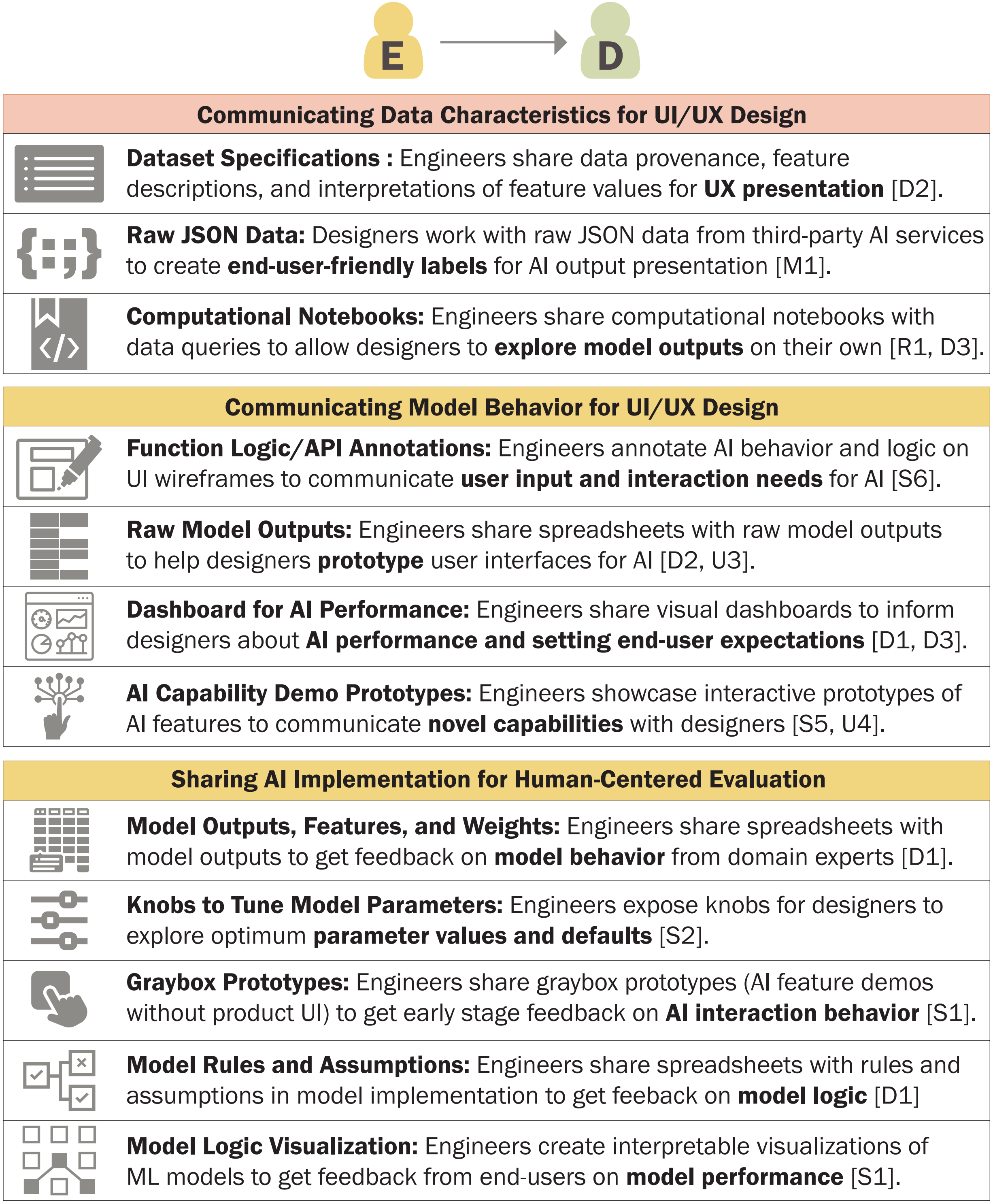
Figure 3.4: Leaky Abstractions Share UX Knowledge to Inform Engineering Decisions

tion leaks allow engineers to (1) communicate about training data characteristics for user interface design, (2) communicate model behavior for user experience design, and (3) un-box AI for evaluation with end-users and designers. For instance, to allow designers to explore training data characteristics, engineers created and shared computational notebooks with ready-to-run data queries along with data specification documents. Access to these details supports designers in determining appropriate interface controls and presentation features such as formatting and categorizing AI outputs. 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. Lastly, accessible representations of AI logic, including interpretable visualizations, spreadsheets with model rules, and controls for tuning model parameters, allow designers and end-users to validate and provide feedback on detailed AI implementation.

By adopting information-sharing practices atypical in conventional software design, both designers and engineers overcome knowledge blindness about technology and end-users. These accessible representations support teams as they collaborate to verify that AI implementation decisions align with user experience design, and that human needs are reflected in AI subcomponents and training data decisions. Leaky abstractions allow designers and engineers to bridge the *implementation* hierarchy covering the product's function, specific implementation logic, and aggregation (part-whole) hierarchy representing how each component fits within the AIX experience. Through abstract information leaks, teams operationalize the HAI guidelines for explainability, error handling, feedback, and learnability. Given AI's uncertainties, leaky abstractions are a necessary feature for accomplishing AIX design.

3.4.2 Delayed Specifications Reduce Friction during Collaboration

Our findings show that premature specifications introduce friction at the AI-UX boundary, making it challenging to implement HAI guidelines. What works instead is co-design by designers and engineers to devise design specifications through the sharing of leaky abstractions. As shown in Figure 3.6, the approach observed is to operationalizing HAI guidelines as *delayed* specifications through iterative prototyping and constant evaluation in order to realize collaboratively defined complete specifications. In the early design stages, designers and engineers produce fuzzy design specifications with some aspects more concretely defined. By sharing those initial design artifacts, 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 de-



Communicating Data Characteristics for UI/UX Design

Dataset Specifications : Engineers share data provenance, feature descriptions, and interpretations of feature values for **UX presentation** [D2].

Raw JSON Data: Designers work with raw JSON data from third-party AI services to create **end-user-friendly labels** for AI output presentation [M1].

Computational Notebooks: Engineers share computational notebooks with data queries to allow designers to **explore model outputs** on their own [R1, D3].

Communicating Model Behavior for UI/UX Design

Function Logic/API Annotations: Engineers annotate AI behavior and logic on UI wireframes to communicate **user input and interaction needs** for AI [S6].

Raw Model Outputs: Engineers share spreadsheets with raw model outputs to help designers **prototype** user interfaces for AI [D2, U3].

Dashboard for AI Performance: Engineers share visual dashboards to inform designers about **AI performance and setting end-user expectations** [D1, D3].

AI Capability Demo Prototypes: Engineers showcase interactive prototypes of AI features to communicate **novel capabilities** with designers [S5, U4].

Sharing AI Implementation for Human-Centered Evaluation

Model Outputs, Features, and Weights: Engineers share spreadsheets with model outputs to get feedback on **model behavior** from domain experts [D1].

Knobs to Tune Model Parameters: Engineers expose knobs for designers to explore optimum **parameter values and defaults** [S2].

Graybox Prototypes: Engineers share graybox prototypes (AI feature demos without product UI) to get early stage feedback on **AI interaction behavior** [S1].

Model Rules and Assumptions: Engineers share spreadsheets with rules and assumptions in model implementation to get feeback on **model logic** [D1]

Model Logic Visualization: Engineers create interpretable visualizations of ML models to get feedback from end-users on **model performance** [S1].

Figure 3.5: Leaky Abstractions Share AI Knowledge to Inform UX Decisions

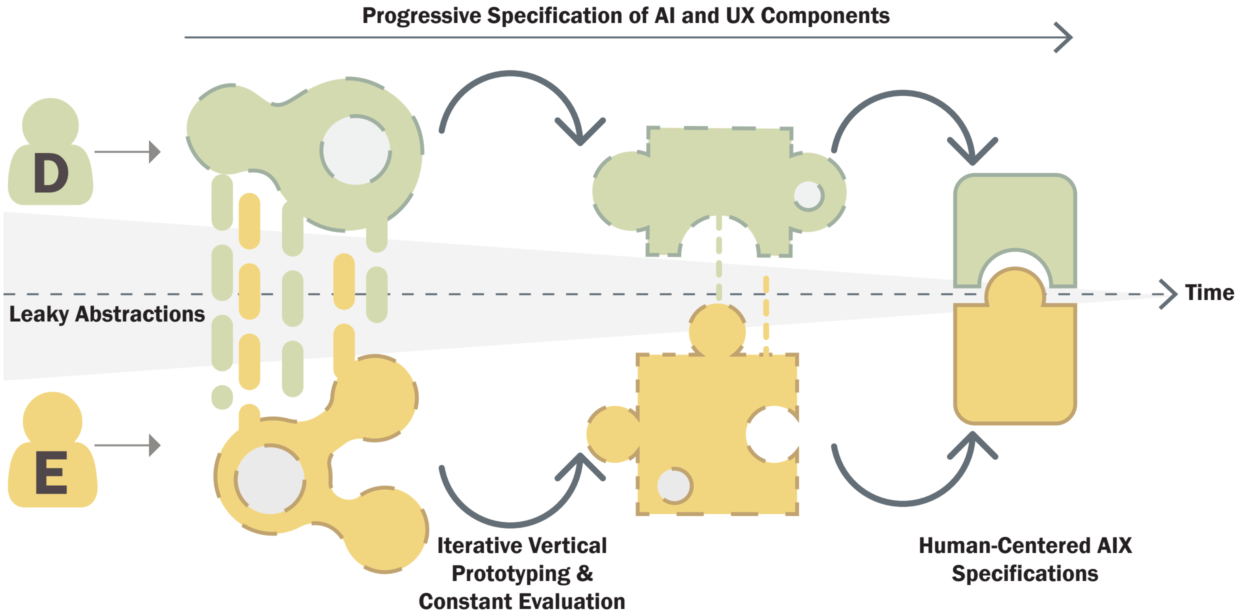

Progressive Specification of AI and UX Components
D
E
Leaky Abstractions
Time
Iterative Vertical Prototyping & Constant Evaluation
Human-Centered AIX Specifications

Figure 3.6: Delayed Specification through Vertical Prototyping and Constant Evaluation

sign of AI. During this stage, avoiding commitment to specifications makes the design pliable and invites collaboration and inputs. As the design progresses, more and more aspects of AI and UX components become concrete, and consequently, the need for leakiness at the boundary reduces. In the final design stages, teams arrive at realized designs *solutions* aligned across AI, UX, and humans.

3.4.3 Coordination to Collaboration and the Need for Co-Design

In conventional software design, a clean separation between UX design and software implementation provides effective coordination and hand-off between designers and engineers. However, there is no clean way to "slice" (or separate) system components and tasks between designers and AI engineers in AIX design. As our findings show, boundaries introduce friction and frustration for both designers and engineers. Instead, successful teams in our study adopted a collaborative approach to AIX where they share emerging design needs and specifications with each other to arrive at communal design solutions. While effective, this approach was observed to be primarily asynchronous in our study, which can be inefficient in the early stages of design. While working separately, engineers envision AI capabilities through minimum viable data and developing demos, while designers offer quick idea iteration and explore alternatives as low-cost, low-fidelity prototypes. For AIX design, it may be possible for designers and engineers to engage in generative design thinking collaboratively in the early stages of design with additional benefits. A possible alternative process would be to extend current generative design practices to include engineers in the co-design of AIX alternatives. Co-design occurring in the early stages of HAI design may help designers and engineers arrive at initial specifications more quickly and efficiently and support further collaboration and feedback as observed in this study.

3.5 Summary

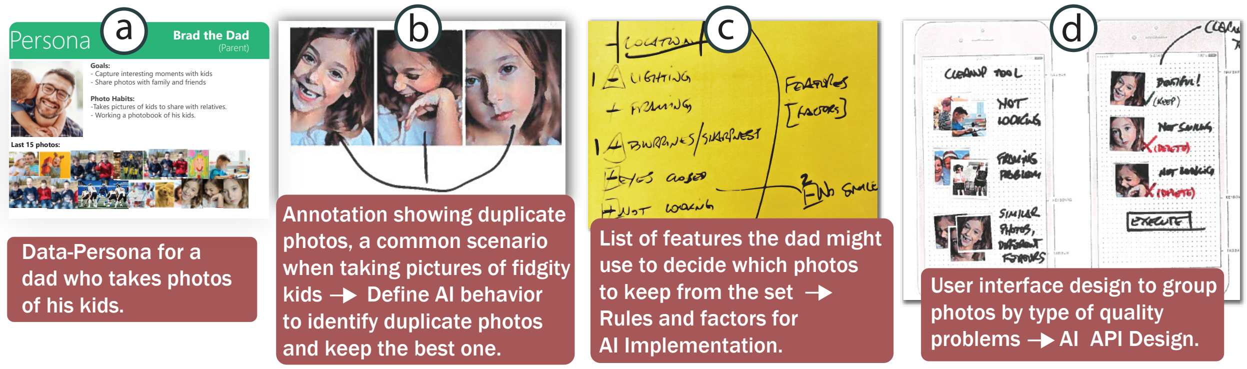
In conventional software development, the boundary between UX and engineering is well defined: designers research and design based on end-user needs; engineers build to those specifications. However, AI application design poses a challenge to this model of coordination. Our analysis of HAI design guidelines (and the resulting AIX component model) shows that implementing the guidelines will require multidisciplinary expertise and collaboration. Based on our interviews with UX researchers, AI engineers, data scientists, and project managers, we identified sets of common challenges to collaboration. Boundaries between designers and engineers introduce knowledge blindness about end-users and technology. For example, designers may not know the possibilities and limits of AI or be equipped to design for AI uncertainties. Engineers describe difficulties in aligning data and AI models with end-user needs in the presence of uncertainty. Further, the data-intensive approach of AI challenges conventional UX design practices. As a solution, we identified that leaky abstractions allow designers and engineers to overcome knowledge blindness and engage in collaborative design. While our interviews surfaced numerous instances of boundary representations that embody abstraction leaks, due to legal constraints our participants were unable to share specific details about their collaboration using leaky artifacts. We did not get to see the artifacts first hand or get detailed insights about how these artifacts supported AIX design. In the next chapter we conduct a focused in-lab study to observe design-engineering collaboration for AIX design.

CHAPTER 4

A Process Model for Co-Creating AI Experiences

This chapter considers the material approach to AIX design by factoring material creation in the design process. When working with new and unfamiliar technology, designers are encouraged to consider it from a "material" perspective [88, 202, 240]. Just as with wood or fabric, in which the craftsman needs to understand the material to create with it, designers need to know what the technology is capable of, what its limitations are, and what properties are available for design. For instance, when working with Radio Frequency ID (RFID) technology, the designer should first explore its material properties including the signal strength, how much information an RF tag can hold, and how quickly the information can be read [12]. This will allow the designer to *manipulate* its properties in creative ways to generate design solutions [24, 55, 213]. However, unlike other technology materials that are created *before* the user experience (UX) design, artificial intelligence (AI) does not lend itself well to a purely material-driven design approach [138, 254]. Instead, AI's material properties only emerge through its application experience design [68, 139]. As a simple example, to design an intelligent To-Do List application that automatically creates tasks from emails (e.g., [82]), designers cannot work with AI as a given material that makes predictions from text. To create the AI material, AI engineers need guidance from designers about how end-users think about tasks, who the potential end-users are, what emails mean to users, and so on (i.e., human-centered AI [15]). For both the designer and the AI engineer, this is a challenging *chicken-and-egg* problem [190, 209]. To address this, we investigate what a co-creation process for AI's *form* and *function* might look like, what AI as a material "under construction" entails, and how the evolving UX design can inform AI development.

A fundamental assumption of the material view in HCI is that materials are a given, and they *possess* specific properties that are amenable to design. To a large extent, this assumption holds. From an engineering standpoint, materials are invented with specific *structure-property* relationships in mind (e.g., [133, 215, 228]). They can be used in any context in which those relationships are desirable [4, 108, 185]. The designer's job is then to explore the material, understand how end-users might experience it, and thereby acquire knowledge for generative design thinking [88]. For instance, to prototype a To-Do List in mixed-reality (a novel material), the designer can begin



Data-Persona for a dad who takes photos of his kids.

Annotation showing duplicate photos, a common scenario when taking pictures of fidgity kids → Define AI behavior to identify duplicate photos and keep the best one.

List of features the dad might use to decide which photos to keep from the set → Rules and factors for AI Implementation.

User interface design to group photos by type of quality problems → AI API Design.

Figure 4.1: User Data as a probe to design AI material: (a) The designer uses data-persona to construct (b) a scenario about a parent taking duplicate photos (designs AI behavior), (c) listing the features the parent might use to identify bad photos (defines how AI should implement the behavior), and (d) creating AI-Powered UI for de-cluttering photo albums (defines inputs and outputs to AI).

by exploring how graphics are rendered spatially, what the visual field of view entails, and which hand gestures are available for interactivity (e.g., [182]). Based on this, designers can prototype alternatives by changing the appearance of graphical elements [174], exploring different gestures and layout options to design the UX. In other words, design with material is accomplished by knowing its *created* properties. Even in extreme cases of customization [154], the design material metaphor holds.

In this vein, AI is also a novel design material [117, 250]. However, there are important distinctions that make it challenging to put this material perspective into practice. First, as a given (or prefabricated) material, AI is *deficient* for design. AI materials are commonly described in abstractions such as techniques (e.g., supervised-learning) and behavior (e.g., prediction), and divorced from contexts in which the AI is applied. Unlike the mixed-reality example, a designer cannot simply explore a "supervised learning" AI to design an intelligent To-Do List. Designing with AI requires *defining* its material properties, including what the AI system should learn using what data, which assumptions and learning rules are appropriate, and how those capabilities should manifest in designed experiences (e.g., data labels). Second, once created by AI engineers, an AI system's properties cannot be readily manipulated during the application's design process. In comparison to mixed reality interfaces, in which the designer can change properties (such as color or shape of elements), representational and knowledge barriers prevent designers from directly altering AI to mold it into a 'designed' product (e.g., [251]). Third, in many AI systems, its material characteristics can continue to evolve through feedback and learning. The designers must anticipate how the AI will change over time and experience. These capabilities require design across both the application and the AI material.

HAI guidelines emphasize the designer's responsibility to understand the AI design material, but not the role of AI practitioners in AIX. The material design approach assumes the AI, like natural wood, must be taken as given; so, the designer provides the required adaptation. For designers, this challenges their technical expertise and introduces friction to material exploration [253]. For instance, designers cannot prototype the user experience with a "fail fast, fail often [251]" approach. With AI material, *vertical* end-to-end prototyping is required to create, evaluate, and revise design alternatives. Such a process is time-consuming for designers and engineers, it is resource-intensive, and the chicken-and-egg problem remains [139]. This is the primary motivation for our work: *How might designers and engineers co-create AI Experiences through rapid, collaborative design?*

Drawing from prior research in HCI and AI application design, we developed a protocol for co-creating AIX through generative design thinking. Using the protocol, pairs of UX designers and AI engineers worked to design AI material characteristics and the user experience for a given design problem. we observed UX designers' involvement in AI material creation, including defining AI behavior, specifying the AI architecture, and features related to explainability, failure, and learnability. we found that end-user data played a critical role in shaping AIX. As shown in Figure 4.1, by using data as a design probe, designers constructed AI-infused scenarios to co-design desired AI behavior. By imagining mental-models for different personas across scenarios and data (i.e., how might the persona perform a task?), they offered inputs to AI architecture design. Through user interface prototyping with data, participants also co-designed the application programming interface (API). Teams created data probes as a scaffold for divergent design thinking, material testing, and design validation. The key contributions of this chapter includes (1) identifying the role of designerly proxies[1] and data probes in defining AIX material, (2) describing a process model for co-creating AIX, and (3) highlighting a set of design considerations for incorporating data probes in AIX design tools.

4.1 Background on Material Design for AIX

A material framework for design includes (1) *fabrication*—ways to produce materials with specific properties, (2) *application*—ways to transform materials into products, and (3) *appreciation*—reception of material by the end-users [67]. Design requires iteration and feedback across these three aspects. When fabrication and application are cleanly separated—as with natural materials like wood and technological materials like RFID—prior work has considered design material as a *given* [12, 65, 79, 88, 98, 184, 148, 206]. As a consequence, UX design emphasizes understanding material *properties*, developing *processes* to generate material artifacts, and evaluating *expecta-*

[1] Designerly proxies are the designer's representations of AI's technical characteristics.

tions and *values* associated with material encounters. Other work has combined material creation (fabrication) into the design problem and investigated *co-creation* of the material along with its application [31, 137, 157, 213, 198, 234]. We draw from both of these perspectives to develop our understanding of designing the AI material and designing with it. By characterizing 'design' as an activity that applies a value system to create objects of reasoning [31], with AI as design material, we identify gaps in guidelines, methods, and representations. Through this discussion, we formulate the research questions for our study.

4.1.1 Guidelines

Numerous design guidelines for AI applications have emerged from both academic and industry research. The guidelines span across functionality [128], end-user interactions [10, 92, 107], learnability [90], explainability [239], privacy [97, 134], transparency [74], etc. Several guidelines address the intersections of both fabrication and application design (i.e., AIX). In some cases, the guidelines ask that we consider application context when creating AI capabilities; For instance, PAIR [92] recommends modeling AI after the human expert: *"When designing automation, we should consider how a theoretical human 'expert' might perform the task today"*. Others offer suggestions for repairing AI material flaws through UX enhancements: *"Make it easy to edit, refine, or recover when the AI system is wrong [10]."* This highlights the inherent dependencies between fabrication and application design for AI [68, 253]. In our work, we look at how designers and AI engineers conceptualize the guidelines from different "points of view" to co-create AIX. To aid our investigation, we consider the language of material engineering which provides a vocabulary for material as *structure*, *surface*, and *properties* [234]. This would allow us to establish connections between material characteristics and AI material experience in its embodiment, encounters, and collaborations [88].

Research Question 4: *How might designers and AI engineers conceptualize shared, and differing, design perspectives arising from human-AI guidelines to co-create AIX?*

4.1.2 Design Methods

Current AI development workflows consist of critical design-related steps, including identifying model requirements, data labeling, feature engineering, etc. [7]. However, in many cases, UX design and AI development only converge after AI decisions have been made [68]. Consequently, UX designers face challenges in incorporating AI material properties within their design practices [250]. Similarly, engineers find it challenging to obtain ground truth validation for AI-related decisions [114], avoid blind-spots threatening responsible AI needs [119], and incorporate necessary UX inputs for improving model performance [231]. Hence, design methodologies should be symbiotic to produce the best application performance. If AI is meant to replicate human intelligence, UX designers can offer insights to make it practically and emotionally resonant with users [46, 249]. The main challenges to collaboration have been time-related constraints to fabrication and design [252], barriers to immediate feedback [139, 251], and lack of motivation and incentives [56, 139]. Further, AI material challenges conventional prototyping methods because it requires a higher level of commitment and effort to prototype AI applications [251, 253]. We also lack means for UX designers to engage in a "conversation with the materials [250]," and ways for the AI materials to "talk back to the designer [250]."

When considering software code as design material, programming becomes a vital part of the design process and offers necessary "talk-backs" for design [157]. Even in natural materials such as wood, material properties (hardness, grain), and constraints (knots and weak points) offer feedback resulting in design recourse [65]. In co-creating AIX, both designers and engineers require the material and the application experience to respond to each other. Separately, when working with Bluetooth as novel design material, designers generate end-to-end fully working sketches (inspirational bits), allowing them to investigate its properties through form-giving [226]. To co-create AIX, designers and engineers need similar low-cost *vertical-prototyping* strategies to create end-to-end prototypes of the UX and the AI backend [24]. Tools for material prototyping should be accessible (allow developers and designers to think about the material), immediate (support rapid iterative feedback, reflection-in-action, and reflection-on-action), and generative (allow test, probe, and exploration iterations) [98]. Lastly, developers, engineers, and data scientists employ more mechanical, less organic processes focused on application *data* [91]. Consequently, a key task for human-centered designers is promoting user experience data as a bridge between the two fields through a process of translation [91, 111]. We incorporate these perspectives about *data* in developing our study protocol for AIX.

Research Question 5: *How might designers and engineers co-create the design and technical characteristics of AIX?*

4.1.3 Representations

Design requires creating and comparing alternatives to arrive at a final solution [213]. The challenge with AIX is finding intermediate design representations that can serve as a "lingua franca" easily understood by multi-disciplinary teams of designers and engineers [31]. Representations would ideally allow the design concept to be viewed differently based on functional perspectives. For instance, for engineers, data is represented as a set of variables [39]; but in UX, data is associated with end-users and their situated context [109]. Designing across boundaries requires a show and tell ("I will know it when I see it [32]") approach. In [139], active discussions between UX designers and ML researchers around mock-ups helped them avoid miscommunication about model capabilities. These rich representations should provide ways to envision viewpoints and resolve differences, and align design needs through negotiation over intermediate representations including words, sketches, physical mock-ups, charts, etc [35, 125]. At the same time, these representations should be easy to create during rapid prototyping. In prototyping web applications, designers use non-functional proxies to negotiate a design that works for both developers and end-users (e.g., Interfake [121], Apiary [186], etc.). Such 'mocks' can circumvent the need for more programming effort to creating AI material designs. Our study explores mixed-fidelity prototypes [172] that provide high-fidelity representations in some dimensions and low fidelity in others.

Research Question 6: *What types of material and design representations can support co-creating AIX?*

4.2 Method

We conducted an in-lab design study in which UX designers paired with AI engineers worked together to co-create AIX (a total of 10 sessions, each with one designer and one engineer). To model the nature of collaboration, we took inspiration from Wizard-of-Oz (WoZ) techniques for AI prototyping [171, 235, 34]. We imagined that designers and engineers would implicitly play the 'wizard' (experts) role during co-creation. This allows for rapid feedback about both the AI material being created by engineers and application experience being prototyped by the designer. Therefore, we recruited participants who had prior experience in AI application domains and working in collaborative teams; The designers in our study had an average of 3.4 years of experience (SD = 2.8), and AI engineers had 3.9 years of experience on average (SD = 2.1). Participants comprised of industry practitioners as well as graduate students with prior work experience (Table 4.1). Participants were paired based on their availability. Each session lasted 2.5 hrs, and we compensated participants with $40 for their time. All sessions were video-recorded. We collected all artifacts generated by the participants for our analysis.

4.2.1 Study Protocol

To develop the protocol, we started with human-AI (HAI) design guidelines, developed by companies, for designers and engineers [10, 92, 128, 192]. Ideally, these guidelines represent 'best-practice' advice that is derived from successful practices within the companies. They offer a starting point to explore how UX and AI roles might collaborate in designing the AI experience. We categorized the guidelines into seven steps spanning AI creation, UX design, and AI-UX design processes (see Figure 4.2). Our steps roughly followed the material design process [138] of

Session ID	UX Designer	AI Engineer
1	4 months	1 years & 4 months
2	3 months	6 years
3	4 years & 2 months	2 years & 1 month
4	7 years & 5 months	2 years & 9 months
5	4 years & 6 months	1 years & 6 months
6	3 months	7 years
7	4 years & 5 months	2 years
8	6 years & 2 months	4 years
9	5 years & 7 months	5 years & 6 months
10	3 months	6 years & 5 months

Table 4.1: Participant details indicating years of experience for designers and engineers in our study.

proposing material, envisioning material experience, manifesting material experience patterns, and making material product concept designs. In addition, we included material creation in the process. In our instructions to participants, we refrained from using the material metaphor; instead, we worked with terminology specified in the HAI guidelines. Further, we organized the steps into two phases; The first phase aimed at producing initial AI specifications (fabrication) and prototypes of the AI-powered UI (application). This included opportunity spotting, model specification, and UI prototyping. In the second phase, participants iterated over the design to arrive at a 'pragmatic' solution by considering errors, explainability, feedback, and expectation-setting for end-users (appreciation). Our first session served as a pilot informing the two phases, and we used the feedback to revise the protocol for the remaining sessions. The second phase would allow teams to consider AI's uncertainties and offer adaptations to account for AI errors, thereby maximizing AI's utility for end-users. At each step of the protocol, the study coordinator offered instructions and tools for participants to work on that step. Each step had a time limit, and participants shared and discussed their design with the coordinator throughout the study.

Design Problem Briefing: We selected the problem of *decluttering the photo album* on the phone. We motivated the problem by stating that cameras on smartphones have made it easy to take photos anytime and anywhere. A consequence is that users capture hundreds of photos that may be of little value. Deleting unwanted photos can be tedious and boring. We asked participants to design an AI-powered solution to address this problem. This domain is simple enough for participants to understand, and they can leverage experiential knowledge in brainstorming solutions. In other words, the problem minimizes domain complexities while allowing us to observe collaboration in a single task session. We provided participants with screenshots of the current (non-AI) photo album interface that we created. In our pilot, we observed that the designer spent time creating the same interface by looking at their phone. Providing this design upfront allows us to focus time on more critical steps. We also specified what the back-end delete API looks like (a simple function that takes a list of photo ids to delete and returns a success or failure message).

Like Zhou et al. [254], we offered participants a set of initial persona cards, including a parent, a business traveler, a 3D-artist, and an Instagram influencer. For each persona, we listed their goals and photo-taking habits. Motivated by prior work in data-driven design [111] and parallel work in data visualization design (i.e., “data changed everything [238]”), we included a set of 15 most recent photos taken by each of the personas. Our data-personas align with the “minimum-viable-data [235]” concept that AI engineers typically use in prototyping machine learning models. From a UX standpoint, the data-driven persona is similar to what designers could generate through user research with mixed-method data (e.g., [222]). We carefully curated the photos to include a variety of images that represented a diverse set of photo capturing behavior and photo content. This was done to ensure a diverse set of AIX solutions.

Step 1—Opportunity Spotting (∼25 minutes): We asked participants to brainstorm ways in which AI can support the decluttering tasks by aligning AI needs with human needs [92]. We provided them with information about types of AI (predictive, perceptual, generative) [192]. We also gave them guidelines about integrating AI experiences into end-user task workflows (e.g., when the AI should automatically perform a task, and when it should take an assistive role when explicitly invoked by end users) [128]. Participants had access to note pads, sticky notes, and colored markers throughout the session to brainstorm. They were also free to annotate on any of the printed study materials. At the end of this step, participants converged on the AI capabilities they would design in the next steps.

Step 2—Model Specification (∼20 minutes): In this step, participants continued brainstorming ways in which they would implement the AI capabilities that emerged in step 1. We provided them with an ML model design template to brainstorm about training data needs and factors they would consider for implementing the behavior [192]. This forced participants to externalize their thoughts and collaborate. We also provided them with printed spreadsheets with persona images on one column and empty columns to fill out with feature values. This encouraged a WoZ like approach to simulate model predictions. They were free to use it as a worksheet to iterate on the design.

Step 3—Vertical Prototyping (∼30 minutes): At this point, we instructed the UX designers to prototype the user interface design using output from steps 1 and 2. We provided them with printed templates for wireframing mobile interfaces. We asked the designers not to use placeholders for text or images in their prototype. We provided them with printed images for each persona (both thumbnail size and screen-size images). Participants could cut and glue the images onto their prototypes (i.e., make medium-fidelity prototypes [75]). We intended to see how participants worked with factual data when designing the interface. For text and labels, when they were unclear what the content needs to be, they were asked to annotate with a question mark for later discussion with the engineer. In parallel, we asked the AI practitioner to fill out model API cards (one for each AI capability), providing details about API name, model inputs, model outputs, behavior description, and details about the training data (adapted from [175]). This comprised a low-cost realization of the AI "material" for feedback and iteration. At the end of this step, they each explained the model API design and the UI design to each other and the study coordinator. We provided participants with translucent sheets (vellum paper) to place on top of the prototypes to annotate and discuss. The goal was to map user interactions to API calls and align model inputs and outputs to the prototype. During this stage, engineers revised the model details through negotiation, and designers updated the interface when required.

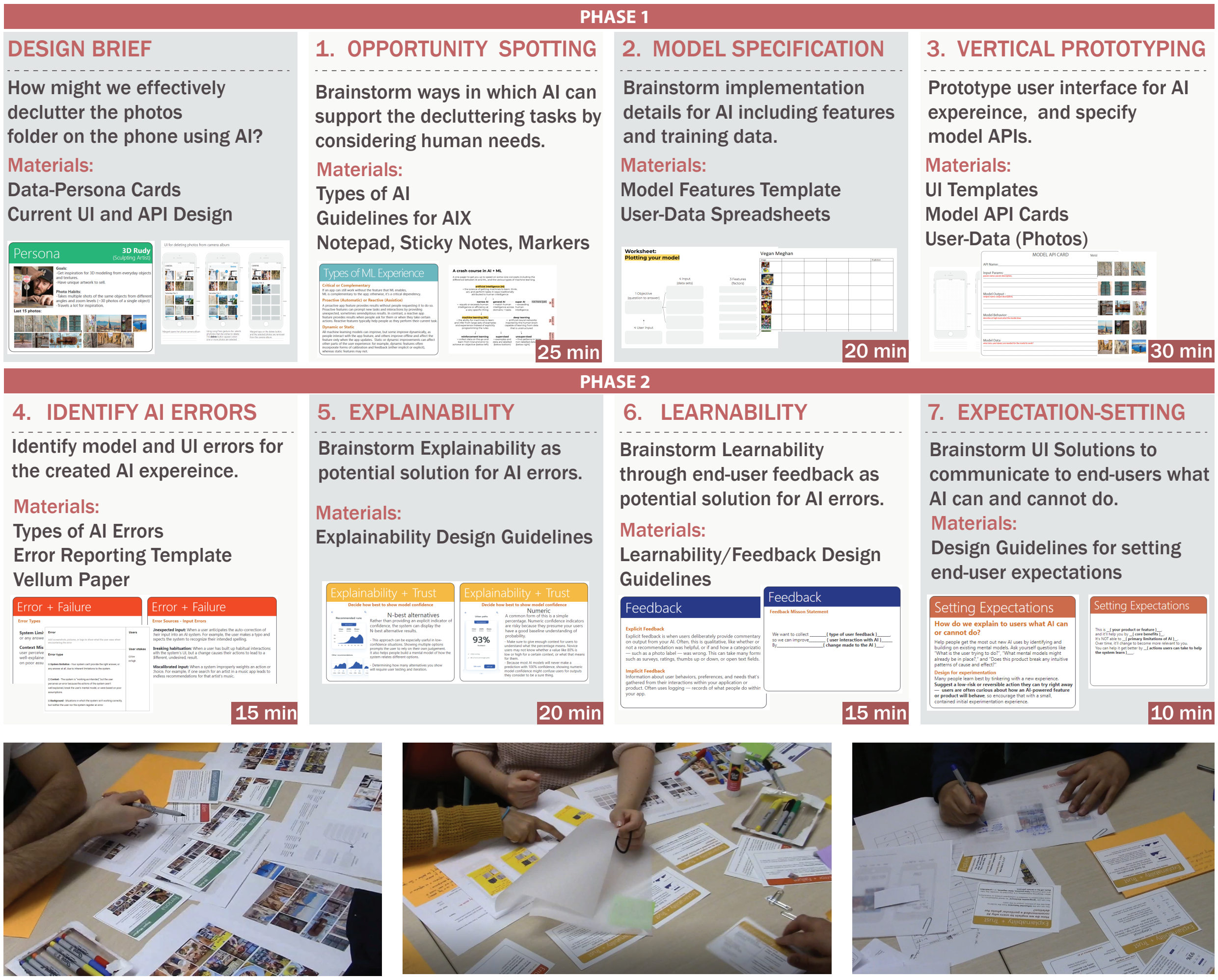
Step 4—Identify AI Errors (∼15 minutes): We created design cards explaining different types of AI errors and potential sources of errors [92, 10]. Using this information and the prototypes (UI and Model cards), we asked participants to brainstorm AI and UI specific errors for their design. We provided then with a template to document errors along with different categories (system limitation, context error, background error [92]), but they were free to use the notepad. For this step, participants had to generate a set of potential errors for their AIX design.

Step 5—Design for Explainability (∼20 minutes): In this step, we instructed participants to consider explainability as a solution to the errors generated in the previous step. According to guidelines, context errors are a type of AIX errors in which the system is working as intended, but the user might perceive an error due to lack of understanding, or mismatch with their own mental model [92]. We asked participants to incorporate explainability into their design (both interface prototype and model API) to resolve AI context errors. We provided them with six design cards listing techniques and examples for designing explainable interfaces [92]. We also provided participants with vellum sheets which they could use to annotate over the prototype to design explainable solutions collaboratively.

Step 6—Design for Learnability and Feedback (∼15 minutes): We asked participants to consider learnability and end-user feedback to improve the model performance. Participants had to design ways to elicit feedback from the users. The key here was to design feedback in a way that can be used for model improvement. We provided participants with information about the types of feedback and guidelines for designing explicit feedback [92].

Step 7—Setting Expectations for End-Users (∼10 minutes): In this final step, participants had to design ways to communicate AI capabilities to end-users. We asked them to consider how they might design for end-user trust and how they might design to support end-user control over the data. We provided participants with guidelines about trust and expectation setting [10, 92]. Participants could create new wireframes or annotate over existing ones.

Collectively these steps follow the material design process by considering fabrication, appli-


PHASE 1

DESIGN BRIEF

How might we effectively declutter the photos folder on the phone using AI?

Materials:
Data-Persona Cards
Current UI and API Design

1. OPPORTUNITY SPOTTING

Brainstorm ways in which AI can support the decluttering tasks by considering human needs.

Materials:
Types of AI
Guidelines for AIX
Notepad, Sticky Notes, Markers

25 min

2. MODEL SPECIFICATION

Brainstorm implementation details for AI including features and training data.

Materials:
Model Features Template
User-Data Spreadsheets

20 min

3. VERTICAL PROTOTYPING

Prototype user interface for AI expereince, and specify model APIs.

Materials:
UI Templates
Model API Cards
User-Data (Photos)

30 min

PHASE 2

4. IDENTIFY AI ERRORS

Identify model and UI errors for the created AI expereince.

Materials:
Types of AI Errors
Error Reporting Template
Vellum Paper

15 min

5. EXPLAINABILITY

Brainstorm Explainability as potential solution for AI errors.

Materials:
Explainability Design Guidelines

20 min

6. LEARNABILITY

Brainstorm Learnability through end-user feedback as potential solution for AI errors.

Materials:
Learnability/Feedback Design Guidelines

15 min

7. EXPECTATION-SETTING

Brainstorm UI Solutions to communicate to end-users what AI can and cannot do.

Materials:
Design Guidelines for setting end-user expectations

10 min

Figure 4.2: Overview of our study protocol for co-creating AIX. Top: Design brief and high-level objective for each of the seven steps. Bottom: Visuals from In-Lab sessions.

cation, and appreciation and would allow teams to offer adaptations for AI's uncertainties during the co-creation process. At the end of the study, we debriefed participants about our motivation to investigate AI co-creation process based on HAI guidelines. Participants had the opportunity to ask us questions and provide feedback on the study protocol.

4.2.2 Data Analysis

To prepare the data for analysis, the first author manually transcribed all video recordings. This allowed them to annotate and capture necessary metadata, such as who created the artifacts and how designers and engineers engaged with the study materials. During transcription, they included screen captures of the video to indicate pointing and show-and-tell actions. They also added scanned copies of corresponding artifacts at appropriate points in the transcript. This was done in-line in a word document (one for each session). We then conducted qualitative coding utilizing a combination of deductive and inductive codes [78]. From literature and our protocol steps, we generated an initial set of codes (e.g., AI fabrication, application design, structure, properties, surface, etc.). For instance, we coded AI implementation details as material structure, and discussions about aligning the UI prototypes and model cards as material surface, i.e., the model API. After coding the transcripts using these codes, we carried out the second round of inductive *in-vivo* coding to analyze the data within each category. The generated codes included references to data and types of communication between designers and engineers (knowledge sharing, negotiation, artifact purpose, validation, guidelines, mentions of UI and AI design elements, etc.). After coding, the authors collectively reviewed and discussed the coded transcripts to identify higher-level themes to answer our research questions on co-creating AIX.

4.3 Findings

We asked participants to co-create an experience for decluttering photo albums using AI (in the abstract). By considering human-centered needs, design guidelines, and user-data context, designers approached the AI material in terms of its *experiential* traits. Engineers, who are technically trained, approached defining AI material in terms of *structural* traits, such as learning algorithms, model features, and model architecture. To bridge these differing viewpoints, teams engaged in rich discussions to ascribe material characteristics to the AI, co-create the application experience and evaluate its fit for end-users (i.e., the user experience). In this process, designers concretized their expectations for the AI material through 'designerly' representations, such as scenarios, mental-models, and wireframes. These *shareable* instantiations served as the designers' *proxies* for their desired AI material characteristics. For engineers, these proxies offered human-centered requirements that allowed them to derive the AI material's technical characteristics. We summarize our study findings in terms of (1) *designerly proxies* for articulating AI material needs based on human needs, (2) *data probes* to shape AI material design, and (3) role of *representational artifacts* as realizations of AIX.

4.3.1 Designerly Proxies for Articulating AI Material Needs

4.3.1.1 Material Properties: User Scenarios as a Proxy for Designing AI Behavior

Based on the protocol, teams started the design activity by exploring the intersections of user needs and AI strengths. By looking at the personas and their data (photos), the designers constructed different *scenarios* (user vignettes) to identify reasons behind photo clutter. These scenarios captured varied perspectives, including photo-taking (creation context), photo usage, and photos as mem-

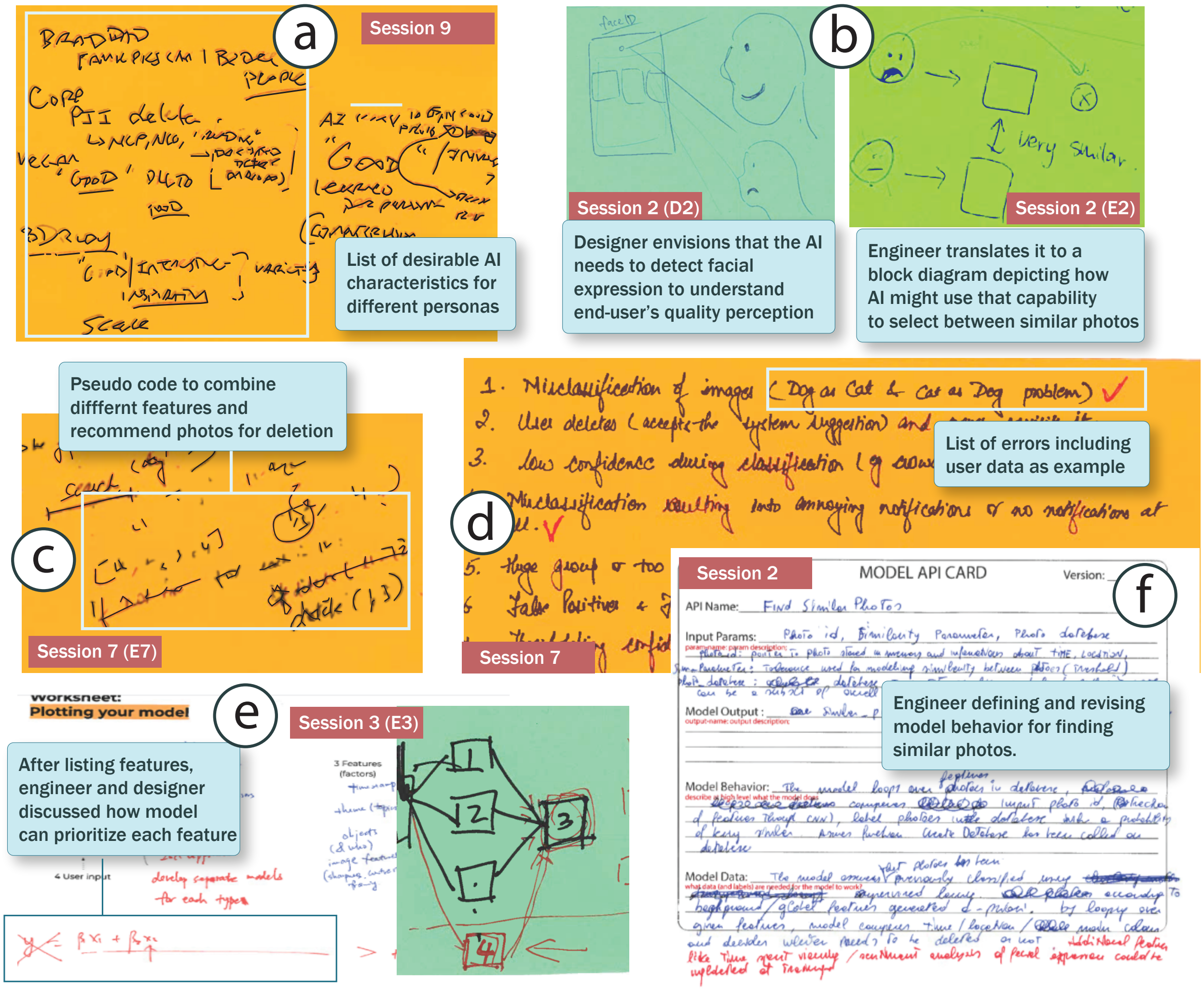

a
Session 9
List of desirable AI characteristics for different personas
b
Session 2 (D2)
Designer envisions that the AI needs to detect facial expression to understand end-user's quality perception
Session 2 (E2)
Engineer translates it to a block diagram depicting how AI might use that capability to select between similar photos
very Similar
c
Pseudo code to combine diffferent features and recommend photos for deletion
Session 7 (E7)
d
1. Misclassification of images (Dog as Cat & Cat as Dog problem)
2. User deletes (accepts the system suggestion) and
3. low confidence during classification
4. Misclassification resulting into annoying notifications or no notifications at all.
5. Huge group or too
6. False Positives
List of errors including user data as example
Session 7
e
Worksheet:
Plotting your model
After listing features, engineer and designer discussed how model can prioritize each feature
Session 3 (E3)
3 Features (factors)
4 User input
develop separate models for each types
1
2
3
4
f
Session 2
MODEL API CARD
Version:
API Name: Find Similar Photos
Input Params: Photo id, Similarity Parameter, Photo database
param-name: param description;
Model Output:
output-name: output description;
Model Behavior:
describe at high level what the model does
Model Data:
what data (and labels) are needed for the model to work?
Engineer defining and revising model behavior for finding similar photos.

Figure 4.3: Designerly proxies and AI material prototypes created by study participants.

ory artifacts (archive). For instance, in session 5, the designer (D5) constructed a scenario where a 'Dad' persona takes a burst of photos to capture his fidgety kids, intending to keep only the best one. Using such scenarios as anchors, the teams then explored how the AI might support decluttering. The designer then asked the engineer (E5) whether the AI could detect similar photos and identify the best one to keep. This was a *conversational* process consisting of "thinking out loud" about different scenarios, supplemented with annotations over the photos (Figure 6.1 b). The designer questioned the engineer about AI capabilities:

> D5: '*'Look at what this Dad is doing, he takes lots of photos of his kids and forgets to delete it, so this is one of the main challenges. Can AI identify duplicate photos and find the best one to keep?"*

> E5: *"Yes, we can cluster the images based on similarity. . . . "*

By constructing such user scenarios (i.e., material application), designers could ascribe potential capabilities for AI in the abstract and co-create the AI's desirable behavioral characteristics (fabrication). Across all sessions, the scenarios led to instantiations of AI with different behaviors, including parsing text information, image quality assessment, and object recognition (Figure 4.3a). Further, in the course of defining AI behavior, designers would revise their initial scenarios to incorporate AI capabilities from engineers, thus creating "AI-infused" scenarios. As an example, D4 added to their vignette that the AI could intervene immediately after the person takes the photo: *"After they take photos you wait until they turn off the phone, and then you have a dialog with the user: 'Hey these photos, the eyes are shut,'... "*(D4).

4.3.1.2 Material Structure: End-User Mental Models as a proxy for Designing AI Implementation

Design would be incomplete without human-centered considerations about the structure of AI material; that is, *How should the AI do what it is supposed to do?* In step 2 of the protocol, we observed that designers approached this issue by simulating in-depth data walkthroughs with previously defined scenarios [194]. During these walkthroughs, the team attempted to construct novel 'mental models' about how end-users might make judgments about decluttering their photos (i.e., which to keep and which ones to delete?), and what the AI can be expected to do for them. We call this an *expectation model* of the end-user. For instance, in session 2, the designer considered the moment immediately after taking a photo and the end user's thought process for deciding whether to take a second (Figure 4.3b). This led to defining *logic* for whether a photo should be deleted: *"... the Dad takes a photo of their kids and then views it for three seconds, which means it might be a good photo or a really, really bad photo and they want to improve it... there is some intention behind it... Can you understand from their facial expression [while looking at the photo] whether this is a good photo?"* (D2).

During these walkthroughs, engineers listened and translated user expectations and decision factors into features, rules, and even pseudo-code for training the model (by writing it down in the model design template - Figure 4.3f). They would question designers about the importance of each feature to end-users and how the model can assign different weights to different features. For example, in session 3, the designer first talked about different attributes of a 'bad photo.' The engineer then visualized a linear model (See Figure 4.3e) to discuss ways to combine those different attributes:

> D3: *"Imagine these four similar pictures, but in this one, I cut his face just a little bit. Can AI identify that? Or if the eyes are closed, and then these are not useful..., and what about lighting or blurriness?"*

> E3: *"From machine learning perspective $y = f(x)$ . . . and each x can be a feature you put into a small model, and we can aggregate the outputs of each small model into a bigger model. . . you can run the same picture through each model and weigh the decision from each model whether to delete or not. . . "*

From these examples, it is apparent that considering both expectation models and associated technical details helped participants co-create the model's structure.

4.3.1.3 Material Surface: User Interface as a Proxy for Designing AI's API

The third aspect of the AI material involves how people interact with it through its surface (i.e., the API). For AI, the API drives the user interface for end-user to engage with AI behavior. In our sessions, designers used the interface prototypes as a proxy when co-creating the model APIs. As with scenarios and walkthroughs, prototyping with concrete data points (as opposed to abstract placeholders such as 'lorem ipsum[2]') allowed designers to articulate specific API-level needs. For example, in session 8, the designer created the interface for viewing a set of recommended photos to delete. By examining this experience, they requested that the API also show key features about *why* the photo was recommended for deletion: *"Let us think about the workflow, it is time to delete, how do you think they are going to process the photos to delete? Are they going to skim it by looking at the thumbnail, or enlarge it to focus on details? Can you provide a smart thumbnail with the features identified by the AI?"* (D8). In their final design, they proposed 'smart-thumbnails' as the AI output. Here, the UI prototypes provided talk-backs for iterating on the material APIs. In most instances, engineers responded by revising the model API card or creating a new API version.

4.3.2 Data Probes to Shape AI Material Design

While designerly proxies offer a medium for articulating AI needs, design also requires generative thinking about alternative solutions. We observed that both the user-data associated with provided personas and participants' 'imagined' data (based on their prior knowledge) facilitated generative design processes. Specifically, participants used individual user-data points as *probes* to construct varied designerly proxies, explore the capabilities and limitations of AI, and evaluate created AI material against different HAI guidelines from the protocol. In human-centered design, design probes promote generative thinking and allow designers to explore the design space [170]. We found that end-user data (e.g., the photos associated with data personas) played the role of design probes in crafting the AI experience.

[2] https://en.wikipedia.org/wiki/Lorem_ipsum

4.3.2.1 Data probes for Divergent Thinking

In constructing user scenarios, participants constantly referred to personas and their photos to brainstorm different AI behavior types. In session 9, referencing the business traveler persona and their photo receipts, the team imagined a natural language understanding behavior to declutter old receipts. As the designer described, *"For this person who is trying to reimburse something, can we delete automatically by reading the text?" (D9)* E9 responds, *"Yes, we can identify text and what is in it, we can use natural language understanding…"* Besides, the data probes allowed participants to think about different ways to implement AI behavior through *mental models and implementation rules*:

> D7: *"Something that came to me when I was looking at the personas was the emotional connections to these pictures like these pictures have values (pointing at pictures of kids). The application has to acknowledge the value for the users and save them instead. How can you classify pictures that have short term values and those that have long term value?"*
>
> E7: *"On the back-end, what I hear is that there are different clusters of pictures, and I understand that different pictures have different value…but there is always a possibility of misclassification"*
>
> D7:*"Could we have overarching rules, like faces might fall into personal attachment bin?"*

Data probes also allowed participants to explore various surface-level features like explainability and end-user feedback to the model. For example, in session 6, the designer did not want to display confidence scores to end-users. To this, the engineer used example data points to illustrate why a lack of explanation may result in distrust for end-users: *"Would the user be displeased if they took four photos and all four had bad lighting, but the system showed the photo with the highest score as this is the best photo because of lighting, but in actuality, they are all bad… Would that cause a loss in trust?" (E6)* Then the designer agreed *"That is a good point, maybe we have 4-5 categories [features], and you show a score underneath for each …, so you know that all four photos are bad." (D6)*. Similarly, in all sessions, participants used data probes to determine the type of feedback that might be useful for model improvements. In session 3, the engineer explained that binary feedback of whether the recommendation was good or bad might not be sufficient for the model to improve. According to the engineer, *"How to use feedback is a difficult task for ML… the thing is users deciding to keep an image may not just be because the prediction is wrong, but because they just like it…for example in this picture the eyes are closed, but she looks cute… One* *way to design the interface is to have more than two buttons; in addition to keep or delete, if there could be a third button, like yes, eyes closed, but I still want it... "(E3).*

4.3.2.2 Data Probes for Exploring Material Boundaries and Limitations

A key aspect of designing AI is understanding the "edge cases," where the AI might fail, or the designed behavior may not work. This understanding is essential for selecting the best alternative at design time (i.e., maximize potential for material appreciation) and designing for uncertainty during use, including failure, error handling, explainability, learnability, and setting expectations. In many instances, designers used data probes to decide whether to incorporate specific AI behaviors into the design. For instance, in session 4, the engineer suggested using smile detection to determine 'goodness' of the photo. To this, the designer pushed back by commenting:*"... if they had a stroke and are not smiling [in the picture], deleting that would be bad... "* (D4). In this case, they incorporated their understanding of the broader domain to think about examples beyond our data personas.

We also observed several instances in which engineers used example data points to highlight the limitations of AI. For instance, E4 commented, *"The model might predict that she is not smiling because of a missing tooth, or braces... "* E5 cautioned about AI limitations on classifying images of kids: *"For adults, facial recognition works well, but I am not sure whether it will work for kids. These all may be photos of the same kids, but I mean kids grow... the clustering may not work."*. These examples helped participants determine whether the user experience with AI should be automated or assistive: *"For duplicate photos, I think it should be assistive and complementary... if the user deletes one photo, we can say here are other photos like this, would you like to delete those as well?"* (E10).

4.3.2.3 Data Probes for Design Evaluation

In addition to exploring material qualities, data probes also supported design convergence and confirmatory evaluation. Similar to identifying edge cases, participants made use of data probes to check whether a behavior 'scaled' across different personas. For example, in session 10, participants identified the desired AI behavior of detecting duplicates and selecting the best photo to keep. The designer then considered an Instagrammer persona for testing whether the same behavior might be useful to them. According to D10: *"Same goes here... after they upload to Instagram they probably do not need it. Especially for Instagram people, they edit their photos a lot, and each time they edit, it will create new copies if it... "*. Similarly, after making decisions about *not* including certain behaviors in their design, teams discussed why that decision was right by considering other personas. For instance, in session 9, they rejected their initial idea about predicting images'

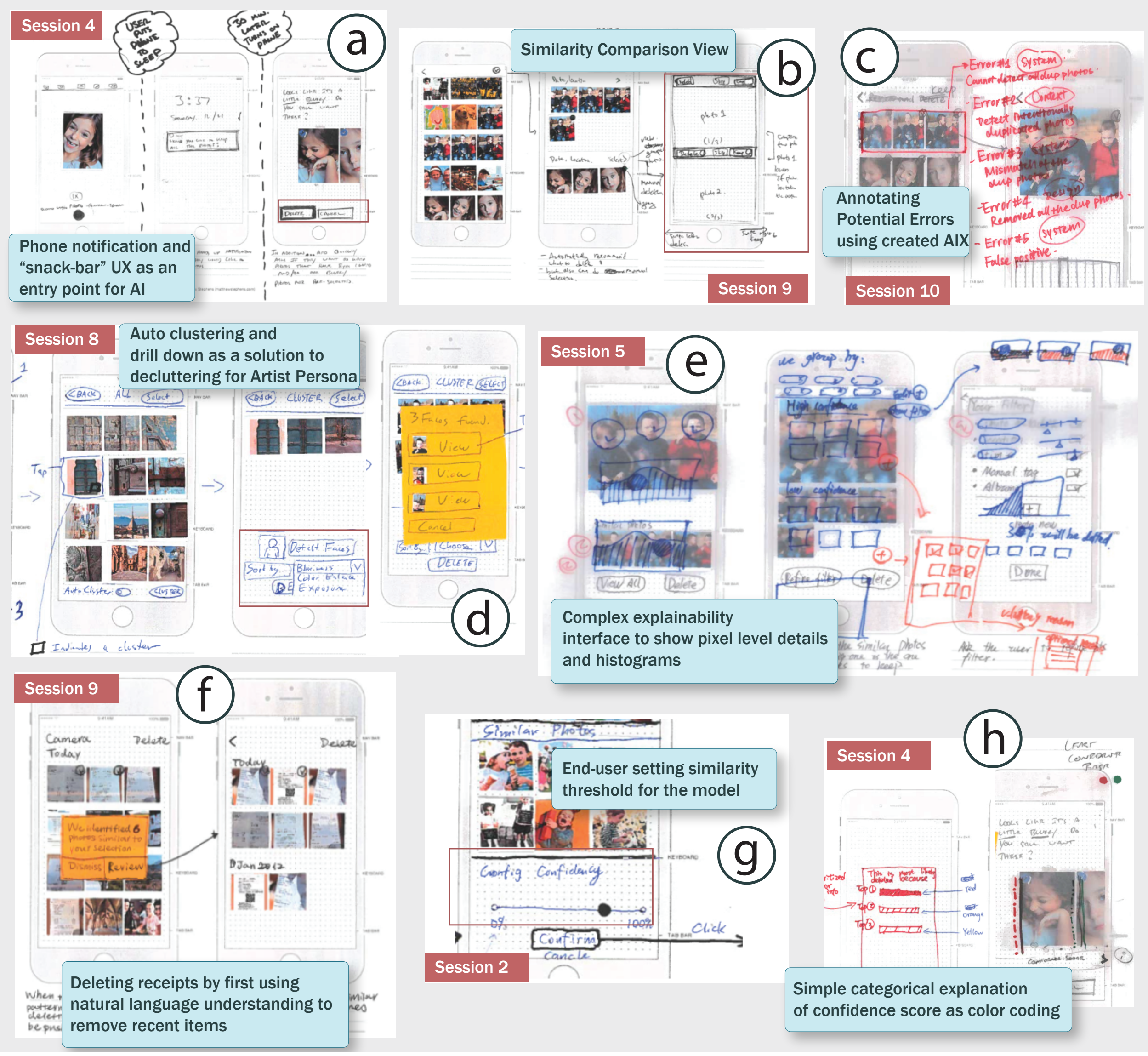

Session 4
a
Phone notification and “snack-bar” UX as an entry point for AI
Similarity Comparison View
b
Session 9
c
Annotating Potential Errors using created AIX
Session 10
Session 8
Auto clustering and drill down as a solution to decluttering for Artist Persona
d
Session 5
e
Complex explainability interface to show pixel level details and histograms
Session 9
f
Deleting receipts by first using natural language understanding to remove recent items
End-user setting similarity threshold for the model
g
Session 2
Session 4
h
Simple categorical explanation of confidence score as color coding

Figure 4.4: AIX Representational Artifacts showing interface design solutions including explainability, and model feedback.

temporal utility. In acknowledging their decision, the engineer E9 commented: *“I am somewhat nervous these [frequency] metrics…for instance, a big failure in which case is deleting a baby’s birth picture, that would be bad…”*

4.3.3 Representational Artifacts as Realizations of the AI Material

Prototyping with user-data allowed designers to experience the AIX ‘first-hand’ as they were creating it—material appreciation. It also allowed them to concretely communicate their use of the created AI material back to the engineers by “representing it to make the solution transparent [213]”.

Given the simplicity of the design brief, we were surprised to see the wealth of design variations generated across the sessions. As shown in Figure 4.4, the entry points for the declutter UX included explicitly invoking the AI via a button click, use of an AI-triggered notification, end-user (delete) actions, providing seed images as a starting point, and conversational dialog. Participants also designed various AI-powered presentation views, including intelligent clustered lists, comparison views highlighting feature differences, interactive thumbnail views, etc. Across all prototypes, the selection of images was driven by imagining AI-infused scenarios and using it to 'play-out' the experience as it was created. Here, we discuss three key benefits to AI material prototyping, including: (1) aligning AI and UI through translation and feedback, (2) addressing misconceptions and gaps in understandings about AI, and (3) helping designers perceive the complex nature of the AI material.

4.3.3.1 Aligning AI and UI through Translation and Feedback:

In constructing AIX prototypes, designers had to translate their understanding of the AI model's structure and APIs into their own knowledge of user interface design. Drawing from their design expertise, they began by representing both the AI behavior and output through familiar design patterns. In many sessions, this provided a starting point for envisioning AIX. In session 2, to recommend to users which photos to delete, the designer started with a familiar photo album interface: *"How to separate good one from all the junk ones? In Apple, they show you all the photos, and you select which ones you want to keep or not..." (D2).* Similarly, in session 4, the designer started with their familiarity about 'snack-bar' UI (i.e., a notification with quick action buttons) to prototype an 'assistive' AI experience: *"...they have what is called a snack bar...where it is a confirmation message, but within the confirmation message you can have a like an undo button, that is really common now..." (D4).*

During the prototyping process, designers concretized their understanding of the model by considering model input and feedback controls within the context of user-data. Their choice of text (such as labels and dialog) corresponded with the AI model implementation rules and features; e.g., in session 4 prototype: "it looks like it's a little blurry, do you want to keep this?". We also observed that designers intuitively designed certain presentation features that led to team discussions and subsequent changes to the model API. In session 4, the designer D4 decided to bin confidence scores into higher-level categories and color code them in photos (Figure 4.4h). Their rationale was to make it more accessible to end-users who may not understand the meaning of differences in confidence scores. This design decision prompted a discussion with the engineer about how the model might categorize confidence scores for a more intuitive presentation:

D4: *"you can give them like some sort of color highlight, so you can give them a*

> *threshold. . . ”*
>
> E4: *“That’s a good point, you can order them by confidence, and you can give them a threshold. . . we can test it out by having a training dataset and test dataset, and we can say from our average, a typical use case the most evenly divided percentages are for example anything below 50 is low, 50 to 75% is medium, anything above is high confidence.”*

The material prototype allowed the team to anticipate user needs and redesign the AI material to address them.

4.3.3.2 Addressing Misconceptions and Gaps in Understanding

Representational prototypes allowed both designers and engineers to identify gaps in each others’ understanding. Through discussions and annotation overlays (using vellum paper), they negotiated differences in their understanding of the AI material. For instance, in session 8, the designer had prototyped detailed text explanations about why a set of images were clustered together. Looking at the prototype, the engineer commented: *“I am not sure technically machines are capable of that. . . people are good at generating semantic explanations, for example people can tell others in natural language why these photos are similar or why they are clustered together. . . But like I am not sure even state of the art ML models are capable of that” (E8).* By contrast, in session 9, the AI engineer suggested using personalized sorting algorithms to present the photos to be deleted. The designer then annotated onto the prototype to clarify that sorting should be objective, and subjective sorting would likely make the user not trust the recommendations.

> E9: *“We can start from a fixed equation and adjust parameters based on user interactions. . . , for example, this image has more exposure than others, then we can personalize the sorting algorithm by changing parameters.”*
>
> D9:*“This is supposed to be an objective way of sorting the value, so if I knew that this sorts blurriness [order photos by blur level] to my choice, I would not trust it anymore.”*

4.3.3.3 Perceiving AI Complexities

The act of constructing an AI material prototype made designers more aware of the complexities of AI-powered interfaces. Across many sessions, by looking at material flaws (e.g., material uncertainties that do not communicate rationale to end-users), designers discovered additional needs, such as setting default model parameters, feedback features, explainability, setting expectations, etc. Working through several iterations on their design, the designer in session 6 commented: *" Now we are 'frankensteining' this sucker, but you can have a settings page that pops open over here and allows them to say, 'keep top N photos'... "*.

While the co-creation process clarified the underlying material structure for designers, some found it hard to separate from their increased knowledge about the AI in order to design the UX, or to communicate their AI understanding to end-users through their design. We observed instances in which the user-to-AI feedback mechanism became very complicated, reflecting the designer's understanding of the AI model but failing to use that understanding in designing the UX. For example, in session 5, the designer and engineer engaged in rich discussions about categorization failures based on pixel-level features and personalization. In applying this new understanding, the designer created a very complicated user interface prototype without considering trade-offs between improving AI model accuracy and the user effort required for feedback (see Figure 4.4e). These examples demonstrate changes in conceptualizations of AI material arising through the co-design of AIX.

4.4 Informal Protocol Evaluation in the Classroom

To gain insights on whether our study protocol can be applied in classroom settings to teach AIX design, we conducted a pilot study in an undergraduate HCI course. A total of 14 students (13 Computer Science Majors, 1 Theatre Major/CS minor) participated in the study. We divided the students into four groups. Based on the class schedule, we administered the protocol over two 90 minute sessions. Prior to the activity, the study coordinator offered an introductory lecture on designing AI-Powered Applications. At the end of the second session, students submitted their design and informal feedback about the activity. At a high level, participants could complete the activity to produce AIX solutions for decluttering photo albums. The designs were similar to what we observed in the in-lab study, but they were not as varied. As CS students, they lacked human-centered design experience, and found it challenging to engage in divergent thinking about the scenarios. As one student commented: *"The most challenging aspect of this design process for me was coming up with unique scenarios that could have been improved with an AI"*. However, students found the vertical-prototyping approach useful: *"I think it was great to see something on paper before it gets into the computer as a code. I really got to think through why certain solutions and why not other solution."*. The design activity helped students understand the importance of co-creating AIX. In reflecting on their session, another student provided an illustration of change in perspective with AIX design:

> *"It seemed as though the biggest difference with this process was that, instead of designing software for a person to interact with, it felt more like we designed a way*

> *for software to interact with a person. The user here is simply meant to respond to what our software does. It sort of flips the script, and leaves our software in control, which is a really interesting difference."*

Encouraged by this feedback, we plan to conduct future sessions with students across CS and HCI disciplines. A pre-requisite to collaboration is familiarity with human-centered design methods and an understanding of AIX design guidelines. Future iterations of our protocol will supply explanations and guided examples to teach students about working with data probes. For instance, methods and frameworks from scenario-based design can offer guided support for generating scenarios with data probes. We also plan to investigate support for students to create their own data probes for domain-specific problems and design goals.

4.5 Discussion

Design materials are central to design processes. In conventional UX design, the graphical user interface (GUI) is the prevalent "design material" that every UI designer understands [44, 79]. If AI were like any other material defined by nature or convention, designers would learn how to work with its given properties to generate design solutions for human users. However, AI resists this approach. Had we provided participants a 'created' AI material (e.g., a closed-box ML model that assigned a quality score to each image [230]), they likely would not have produced the range of expressive, human-centered designs that we observed. Instead, as our findings show, AI material must be defined by investigating the human user's envisioned experiences. In our study, we aimed to answer: (1) How do designers and engineers conceptualize design guidelines from AI and UX perspectives? (2) How do they co-create the design and technical characteristics of AI materials; and (3) What representations are invented during this process? Based on our findings, we respond to these questions by proposing a process model for co-creating AIX and by reflecting on the role of end-user data as a design probe for generative design thinking. Through this discussion, we offer design considerations for data probes within AIX design tools.

4.5.1 Towards a Process Model for Co-Creating AIX

In the study, our teams displayed a design progression occurring across a *spectrum* of materiality. Teams started the design activity by exploring the intersections of user needs and AI strengths. First, with wholly *imaginary and abstract* material, the team worked together to envision the AI. The designers negotiated this initial form-giving by constructing user scenarios, allowing them to approach AI through its potential capabilities without detailed engineering knowledge. This is

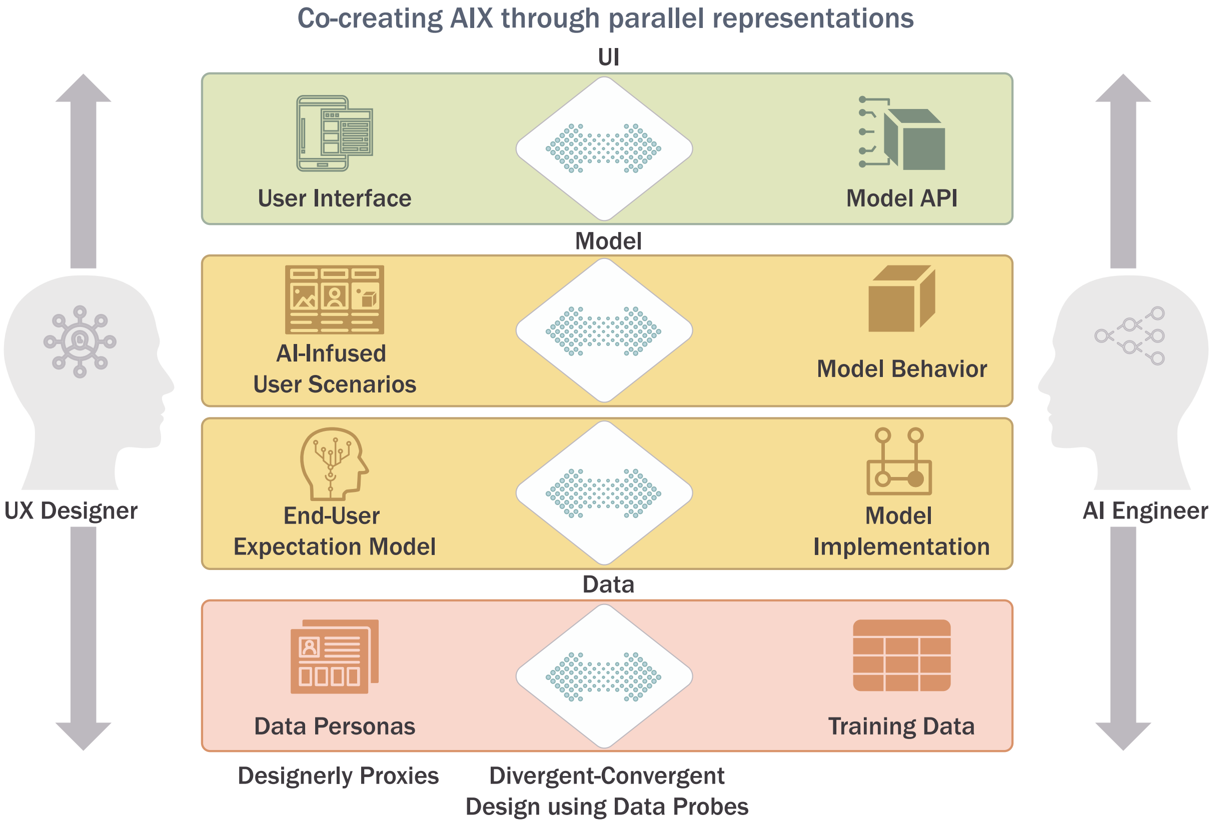

Co-creating AIX through parallel representations
UI
User Interface
Model API
Model
AI-Infused User Scenarios
Model Behavior
End-User Expectation Model
Model Implementation
Data
Data Personas
Training Data
UX Designer
AI Engineer
Designerly Proxies
Divergent-Convergent Design using Data Probes

Figure 4.5: A process model for co-creating AIX.

similar to the role of visualization as proxies in designing ‘immaterial’ materials [12]. Designers used personas, data points, vignettes, and user scenarios—their designerly proxies—to create initial instantiations of the desired AI behavior. These proxies offered design representations as abstractions of *planned behaviors* that allowed engineers to define the technical characteristics of the AI material. In the course of co-designing AI behavior, designers also revised their initial scenarios to incorporate new AI capabilities, thus creating “AI-infused” scenarios.

Equipped with an invented form, teams moved forward to *enact and specify* the AI material. Following human-centered walkthroughs of scenarios, the designers constructed novel *expectation models* capturing how the AI’s end-users might make judgments. The engineers translated these identified user expectations and decision factors into features and rules for training the AI. This co-creation process led to discussions about the attributes, priorities, and values important to users and the technical capabilities the AI needed to support them. While the AI material has taken on behavior and structural characteristics, its envisioned design was only fully *realized* through the team’s use of material prototypes. Designers used interface prototypes as a proxy when co-designing the model’s inputs and outputs (i.e., the material’s surface). These designerly proxies allowed the team to align the AI and UI through translation and feedback. Identifying specific

AI and human behaviors allowed the evaluation of material flaws, misconceptions, and scalability issues. This is similar to *Replay Enactments* [118] that use authentic data to make complex system behavior tangible to designers. Only at this late stage could the full scope of the AI material and UX design be made visible in its interactive complexity.

Clearly, the *dynamic* nature of AI material is unlike other design materials; consequently, the design process for AIX differs from standard design approaches. In conventional human-centered design (HCD), like the double diamond framework [55], the design process is linear or top-down. Designers mainly work at the user interface layer to specify the end-user experience. They then hand-off the created specifications to engineers to build [209]. However, when designing AI experiences, design extends *beyond* the interface and into the design of AI components, including the model's behavior, learning characteristics, assumptions, and nature of training data. UX professionals lack the means to engage in designing these AI components. Instead, current AI development workflows take an "AI-first" approach in which the AI material is created before envisioning its use. Such an approach is problematic because any changes to align the AI's properties to human needs will require costly rework (e.g., addressing disparities in gender classification [40]). Further, there are instances in which the AI behavior itself does not align with human needs, values, and concerns (e.g., using facial recognition to expose political orientation [145]).

In order to address these issues, we need a process in which AI material creation and its application experience design can happen in parallel through iteration and feedback. As shown in Figure 4.5, we propose a process model that combines top-down (UX-first) and bottom-up (AI-first) workflows to distribute agency between designers and engineers. As represented by the *bidirectional* arrows in our model, the AI and UX components are designed in parallel, a critical insight from our study. Our approach shifts engineers' mindsets towards more proactive engagement through accessible user-data proxies and data probes during the co-creation process. Designers engage in co-creating AI behavior without technical roadblocks, operationalize HAI guidelines, and reduce time to feedback (a concern with AI design [251]). Our model's parallel process affords immediate feedback for both material creation and design, obviating the significant rework costs (from collecting and training with new training dataset, retraining the models, etc.) when even small changes arise later. Our study lays groundwork for a collaborative process to align AI's form and its function in the early stages of design. Future research should build on this parallel process model to investigate specific data and representation needs across different application domains and AI capabilities.

4.5.2 Role of Data Probes in AIX Design

Data probes served as a “content common ground” for designers and engineers to collaborate across the different stages in the process model. A characterization of this collaboration is that designers are immediate consumers of AI materials. Their objective is to ensure that the material specification meets their end-users’ UX needs. Using data probes, designers advocated for end-users during the material creation process and simultaneously tested the AI material under construction. Similar to [199], end-user data offered necessary grounding for designers to advocate for centering people in the design of AI, including its behavior, implementation, and APIs.

As shown in Figure 4.5, each of the parallel stages in our model involved both divergent and convergent processes. Designers and engineers ideated on UX needs and AI capabilities together, and they mutually constrained convergence towards a design solution. User data as probes played a critical role in this divergent-convergent process of creating the AI experience. We can extend the material metaphor and borrow from the language of physical material design to characterize the role of data probes: (1) data *molds*, (2) data *vulcanizers*, and (3) data *coupons*. In the early stage of the study, data probes functioned as “molds” for AI’s initial form-giving. By constructing AI-infused scenarios with data probes, designers and engineers explored different forms the AI could potentially take in supporting the declutter experiences. After identifying the initial form, designers used data probes to define the AI material’s internal properties. By constructing expectation models with data probes, designers and engineers “solidified” the AI’s implementation requirements. This step in AI material creation is analogous to ‘vulcanization’ chemicals in the rubber manufacturing process to solidify its internal structure. Finally, by constructing AIX interface representations with real data, designers produced coupons (test samples) of the material to assess the AI experience. In traditional material design processes, ‘coupons’ are samples of the material used to test its properties at a small scale (e.g., [124]). The designers’ mixed-fidelity prototypes served as coupons to test the AI material and address gaps in the desired AI experience.

4.5.3 Design Considerations for AIX Design Tools

Prototyping is an essential step in software development [209]. Through iterative prototyping, teams incorporate increasing details to define different software aspects [24]. The mixed-fidelity approach in our study is an initial step to iterative prototyping. As details increase, teams need to increase the fidelity of their prototypes as well. In this regard, UX design tools should escape the “closed-box” view to make AI more accessible and transparent to designers [252]. Beyond helping designers understand AI (i.e., educational goals), designers should be able to work with AI material during AIX design. The insights from our study suggest that data probes offer useful design considerations for this goal. With this in mind, we offer a set of design considerations for incorporating data probes into design tools.

Support for creating data probes: In the current study, we constructed the data for each persona to include a variety of solution alternatives. Participants also imagined their own additional data points during the design process. Design tools should allow designers to incorporate data from user research into AIX design processes directly. This could include data collected from participants (similar to Wizard-of-Oz prototyping [34]), through mixed methods persona creation (e.g., Data-Assisted Affinity Diagramming [222]), or from dedicated data collection and annotation pipelines [91]. In addition, tools should support accessible ways to generate user data with desired properties. In the data visualization community, tools exist to create datasets with desirable statistical properties (e.g., [94, 168]), allowing designers to select charts to fit their data needs. Our study teams imagined varied data—blurry images, variations in size, and time-progression photos—using their understanding of the task set within use contexts. Design tools should support such expressive 'queries' to find or generate *just-in-time* data probes for designers.

Support for interactive AI & UX design workflows: To work with the AI material under construction, designers can use data probes to receive "talk-back" from the AI design workflow. Currently, end-user machine learning tools such as RunwayML [176] allow novices to interact with machine learning models and for visual exploration of machine learning behavior (e.g., the What-If Tool [93]). However, these tools do not support the use of data probes for divergent thinking about AI behavior. To be effective, probes should be integrated into generative prototyping workflows to provide input and feedback to designers. Moreover, interactions using data probes allow designers to propose desired outputs based on human needs. This can be in the form of ground truth data, annotated labels, output format, etc. For instance, designers can curate a set of diverse data points and ground truth outputs and compare the working model's output against ground-truth values.

Support for constructing designerly proxies: Currently, when prototyping UIs, designers typically work with static placeholder content. As with data visualization, in AI, "data changes everything [238]." Prototyping tools should allow designers to construct AIX design candidate representations by incorporating data probes and AI material talk-backs (e.g., [143]). This allows consideration of alternative choices [162], along with design for AI uncertainty through explainability, learnability, and edge-case analysis[10].

Support for communication during co-creation: A final consideration for co-creating AIX is to share intermediate proxies of AIX, including scenarios, mental models, and interface prototypes with AI engineers. For engineers, they need to offer descriptions of AI properties, assumptions, learning rules, and API details back to designers. This is fundamentally different from standard workflows in which designers primarily share final design specifications with engineers. Both UX design tools and AI creation tools should incorporate features to import, translate, and share intermediate design proxies. Similar to the transparent vellum paper during our study, digital tools should support annotation overlays, generate new examples with different data probes, and communicate failures and constraints (i.e., explainability for designers).

4.5.4 Limitations and Future Work

Our study's design problem was to define an AI-powered experience for decluttering photo albums. While this simple problem allowed us to observe co-creation processes in an accessible domain, other data types may be more challenging for co-creation. More complex design problems may be difficult to represent with pen-and-paper approaches. We plan to conduct co-design sessions with other data types and problem domains to iterate on the process model. We aim to assess (and address) the fit and shortcomings of our process model for different AIX problems through these sessions. Second, we provided participants the data probes for use in our study. We plan to investigate inclusive and participatory approaches to creating diverse data probes for AIX design. Third, our protocol demonstrates a low-cost, rapid prototyping approach to co-creating AIX. As described, this is the first step to iterative prototyping with increasing levels of fidelity. We are currently exploring high-fidelity prototyping tools for AIX to understand how teams might continue to evolve their designs. As an example, we developed ProtoAI [224] to allow designers to invoke models and services with concrete data during prototyping. Fourth, we recruited participants with prior experience in designing AI applications. In many industries, both designers and engineers are new to AI [41]. We are now using the study protocol in classrooms to teach design and engineering students about AIX design. Through these efforts, we will investigate the types of training and scaffolding designers need to effectively participate in AIX's rapid prototyping. Finally, future work should investigate how other stakeholders, including domain experts, representative end-users, and data analysts, might participate in the AIX co-creation process.

4.6 Summary

Treating technology as a design material encourages designers to explore its properties for UX design. However, when working with AI as design material, neither a form-follows-function nor a function-follows-form approach is practical. Instead, the AI material and its application UX need to be co-created through collaboration between designers and AI engineers. In this chapter, we investigated such an approach by conducting an in-lab design study with ten pairs of designers and engineers. Our protocol combines a vertical prototyping approach with talk-backs from AI and UX to facilitate co-creation. We identified the crucial role of end-user data as a tool for co-creating AI design material. By using data probes, designers were able to construct designerly proxies and specify material needs for AI. Data probes facilitated divergent thinking, material testing, and design validation. Based on these findings, we propose a process model for collaborative AIX design and offer considerations for incorporating data probes in AIX design tools. Informed by this process model and design considerations, the following chapters develop design methods and tools for AIX design. Our objective is to extend established design methods using insights from the interviews and in-lab studies to support the data complexities in envisioning and prototyping AI-powered applications.

CHAPTER 5

Data-Assisted Affinity Diagramming

In the in-lab study presented in Chapter 5, *data personas* played a critical design probe in generative and collaborative design thinking. The data personas represent nuanced user segmentation across both qualitative and quantitative views of end-user data. In conventional design practices, affinity Diagrams (AD) and related approaches are the method of choice for clustering data into distinct personas [105]. When conducting AD, designers typically produce physical sticky notes with qualitative data such as interview and observational notes. An advantage is that the notes can be placed on walls or surfaces in a way that leverages spatial cognition, offers flexibility in grouping and clustering, and then physically *persists*. Though software tools have been implemented to emulate and significantly extend the AD experience [87, 241], many designers still favor the traditional, physical, 'sticky-note-on-wall' methodology [99].

While there are numerous advantages to the physical approach, it prevents the adaptation of AD practice for constructing data personas for AIX that incorporates qualitative user characteristics with quantitative data (essential for AI training requirements). Our analysis of prior literature reveled that mixed data analysis also involved data from surveys [33, 61, 102, 123], sensor data [122], and interaction logs [57, 100, 135**?**]. In addition, our pilot interviews with industry practitioners revealed that they often bring their laptops to AD sessions in order to access quantitative data from spreadsheets or summary reports. In their current practice, designers look up quantitative insights that correspond to interview notes (e.g., interaction log data corresponding to "problem controlling music using voice") and make a note of them on the affinity wall (AD notes serve as "magnets for more details"). This approach is not only time consuming, but also problematic in that coherence between the analysis on the wall and the analysis on the screen is hard to maintain. Thus, the motivating question for our work is how we could expand AD for this new type of design process to support the creation of data personas for AIX design?

By conducting a design probe with affinity diagramming users, we identified three main concerns: (1) the affordances of physical notes should be maintained, (2) additional data and insights should be easy to retrieve, and (3) data should be available just-in-time, without disrupting the primary diagramming practice. On this basis, we propose *Affinity Lens*, an augmented reality (AR)

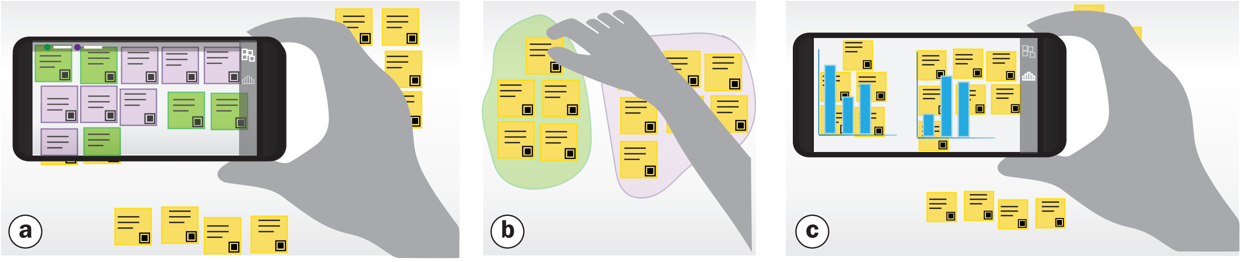


Figure 5.1: ProtoAI used to split a larger affinity cluster based on income level. (a) The user applies a heatmap lens to an existing cluster which shows two sub-groups. (b) The designer regroups the notes. (c) A histogram lens compares sleeping schedules for the two sub-clusters found in (a).

based tool for *Data-Assisted Affinity Diagramming (DAAD)*. Affinity Lens addresses these three concerns by leaving the physical notes in place while using the phone's camera and software to understand the note layout and to 'project' quantitative insights or overlay information on top of the notes and wall surface.

As a simple example, take a designer analyzing comments on a new IoT-based clock radio to determine which features to add. In addition to the text of the comments, the designer also has associated demographic information for each participant. The designer may begin with the comments as affinity notes, ending up with three clusters. The benefit of Affinity Lens becomes apparent when the designer starts looking for deeper patterns. For example, the designer decides to explore the implication of higher level incomes on the kinds of comments from users. By pointing the phone towards a cluster, the designer can easily identify notes from people with high and low incomes and separate them into two different clusters (Figure 5.1a). Once the new clusters are formed (Figure 5.1b), the designer can use the phone to look at distributions of sleeping schedules for each cluster (Figure 5.1c).

Affinity Lens is designed to play an *assistive* role. It allows the designer to maintain their existing (favored) work practice while at the same time offering on-demand analysis. In this sense, the process is backward compatible, both as documentation of an analysis effort and as a usable 'analysis artifact' that can be manipulated beyond the AR. The key contributions of this chapter are identifying where data-assistance can augment AD; implementing a DAAD-focused system, Affinity Lens, which provides an array of extensible AR lenses; and validating, through two studies, that rather than disrupting AD, DAAD and Affinity Lens enriches the practice.

5.1 Related Work

Affinity diagramming (also known as the KJ Method) has been used extensively for over 50 years [208]. AD supports organizing and making sense of unstructured qualitative data through a bottom-up process. A *schema* is developed by individuals, or groups, who arrange and cluster paper notes based on similarity of content, i.e., affinity. Because of its wide use, several projects have worked to address the shortcomings of the basic, 'pen-and-paper' use. These have centered around several areas including remote collaboration, clusters creation assistance, explicit and implicit search mechanisms, general visual analytics systems, and systems to bridge digital and paper documents. We briefly touch upon each area to set the context for the Affinity Lens project.

Collaboration: A number of studies worked to enhance the collaborative nature of affinity diagramming. Though some efforts focused on better-shared spaces (e.g., digital tables [129, 232]), others tackled the individual's role in a shared space by creating different private and shared views (e.g., [241]). These projects seek to enhance the collaborative experience and isolate areas where individual work can happen (likely leading to more diverse observations [70]). With Affinity Lens, we preserve the shared space by maintaining the majority work in the physical space. However, each participant can use their own device to support private analysis (currently we do not synchronize analyses). Affinity Lens can also track changes in the display (indicating what changed since last time) to support both the individual's work over a long period or for asynchronous collaboration.

Cluster creation: Exploration of how people organize information goes back several decades. Malone's early observations on physical organization [165] have been extended and adapted for digital interfaces. Tools for assisting in the creation of clusters have used everything from UI to ML techniques (e.g., [11, 64, 71, 136]). The general idea is that a user should be able to ask what cluster an individual item belongs to, or conversely, what items belong to a chosen cluster. The iVisClustering [152] work provides summaries of clusters including representative keywords and a cluster similarity view. While these have proven useful, the transformation of these object from paper to digital form has limited their widespread use. Though we do offer support for automatic clustering, our focus is enabling the end-user to drive this process. Put another way, Affinity Lens aids the sensemaking process [193] rather than attempting to automate it.

Explicit and Implicit Search: Several projects have explored simple aids for search. These include iCluster [71] and Note Finder [99] which support keyword-based search for matching cards. This capability has been implemented almost directly within Affinity Lens. However, as noted in this past work, this capability is insufficient to be useful on its own. Other efforts have used visual cards as jumping off points for pulling in additional information. Notably, the implicit search work of Dumais and colleagues (e.g., [72]), and the Scatter/Gather work [58] help take affinity diagramming from schematization into additional information gathering.

Visual Analytics Systems: Some prior work explored the notion of a spatial environment for more formal analytical tasks [247]. While completely digital, the notion was that notes could be linked with other notes and augmented with rapid grouping technique and analytical visualizations. The Jigsaw system extends these actions with a greater variety of support for quantitative analytics [220]. We incorporate lightweight, analytic summarizations in a similar style to both of these systems through specific summary lenses. Affinity Lens builds on other, related, visual analytic techniques including the set visualization techniques of [5], where set membership summary information is important to understand overall concepts and the interactive word clouds for summarizing coded text in grounded theory analysis [47].

Paper to digital transformation: Even with these many different directions of work, affinity diagramming in its classic form remains in frequent use due to the extremely low barrier for entry (i.e., sticky notes, pen, and a work surface). In Harboe et al.'s in-depth review of many of these tools [99], they arrive at the same conclusion that we do: instead of trying to replicate paper on screen, tools should offer ways to augment paper notes and support seamless integration between paper and digital worlds (e.g., [130, 131, 142, 149, 178]). The Affinity Note Finder prototype [101] explores one aspect: search. Issues of implementation (slow, heavy device, delay in responsiveness) were an issue, but the biggest concern was that keyword search alone was not sufficient for finding notes. This makes it clear that any single augmentation to the affinity diagramming process must work in conjunction with a constellation of desired activities. Affinity Lens expands that support to include other significant activities in the overall analytics process.

Other projects have explored the paper-digital divide in ways that seek to emulate the large-surface experience of AD. Some sought to bridge the gap by using touch-based interaction on tables and screen. For example, Affinity Table [87] attempts to replicate the look and feel of paper notes by providing natural inking and gestures on a digital display. The iCluster [71] system was implemented on top of a large interactive digital whiteboard. 'The Designer's Outpost' [142] of Klemmer et al. also uses sticky notes and an interactive whiteboard to support the transformation of physical to digital. When a sticky note is placed on to the whiteboard, it is scanned through cameras and subsequently manipulated digitally. The model for Affinity Lens is to preserve the note as a tangible object and virtually augment the information with overlays. That said, to support a number of lenses, Affinity Lens recognizes notes and tracks them in a virtual model.

There are a few additional UI interface metaphors that we build upon. The basic interaction metaphor, that of overlaying additional information and different representations on top of the existing material, draws heavily on the concept of the seminal Toolglass and Magic lens work of Bier et al. [28], as do many other augmented reality experiences. We heavily borrow on overlays and augmentation throughout the Affinity Lens user experience. We also use the concepts from Baudisch et al. [22] for helping give cues to the locations of notes that are currently off-screen.

5.2 A Design Probe for DAAD

To better understand the design space for data-assisted affinity diagramming we initiated an affinity diagramming exercise. The probe had participants work on an artificial task that contained textual comments augmented by associated quantitative data. Participants could also request analyses (in the form of printed visualizations) based on quantitative questions. These were produced by a study administrator who was present in the room with laptop and printer.

We recruited 10 participants who were either UX professionals or HCI-focused graduate students. They all had prior experience with AD, statistics, and data visualization. To encourage participants to think aloud and simulate a more realistic collaborative diagramming session, we had participants work in pairs (5 sessions). Each session lasted 75-90 minutes, and participants were compensated with $20 for their time. The high-level task had participants construct affinity clusters to answer a clustering task. After the subsequent implementation of Affinity Lens, we returned to this task with other groups using the working tool (Section 5.7).

Task and Dataset: We asked participants to analyze a dataset consisting of college students' food choices and cooking preferences using AD. The dataset included: descriptive summaries of a student's current diet, along with other behavioral and demographic attributes including how often they cooked, how often they ate outside, living arrangement, employment, family income, grade point average (GPA), body mass index (BMI), grade level, how often they exercised, marital status, and a self-rated health score on a scale of 1-10 (total of 11 variables) [188]. We selected sixty observations (rows) from the dataset, ensuring that there were plausible clusters in the set that were not too skewed (e.g., 55 people in one, five people in the other). We also ensured that the data varied on different dimensions to encourage the use of a combined analysis approach to form clusters. Each row was printed on a separate note and included an identifier, the text summary, and a table with responses to the 11 variables.

At the start of the study, participants were briefed about AD (though all were familiar with it) and introduced to the dataset and its attributes. They were instructed to cluster the students into six groups (with a maximum of 12 students in each group) such that each group could be assigned to one of six advertisements about food-related services based on their current diet. In addition, participants were provided with summary visualizations for all of the data attributes and were told that they could request additional visualizations on-the-fly based on note IDs. Although visualizations were produced as-requested, the study coordinator kept track of clusters being produced physically on the wall. This ensured that we could quickly generate requested visualizations for notes or clusters. Thus, participants could focus on AD rather than inputting clusters or learning a visualization package.

All sessions were video recorded, and the study coordinator made observational notes and prompted participants with clarifying questions about their clustering choices. At the end of the session, participants provided feedback through interviews. We analyzed the recordings, interviews, and final clusters from all five sessions. Broadly, we found that data-driven insights (i.e., quantitative analysis) supported decisions at all stages of the affinity diagramming workflow. More specifically, data informed a number of task-specific *decision points* for AD. These decision points can be grouped into four main 'assistance' categories: (1) detail access, (2) search, (3) clustering, and (4) summarization. Common AD tasks, such as identifying outliers, were often approached using multiple assistance categories. We provide details and examples for each below.

Detail assistance: A common task in AD is text *interpretation*. From this, topics can be extracted through heuristics to determine affinity. In cases where the text did not provide sufficient details (i.e., lacked clarity) or when interpreting text was hard, participants referred to data attributes to make inferences. For instance, one of the responses in the dataset was *"I eat 3000 - 4000 calories per day and ... "*. Here, participants referred to BMI and exercise levels to disambiguate between an athlete with high caloric needs and someone who might be obese. As a consequence of accessing the quantitative data in relation to clustered subsets, participants began to find novel associations (e.g., responses that mentioned being busy were associated with employment or a living situation; and those who mentioned eating a high protein diet were associated with low BMI and exercise routines).

Search assistance: When a combination of data attributes was *perceived* as anomalous (e.g., a 4th-year student living on campus, or someone who eats healthy but has a low health score, etc.) participants attempted to look for other individuals with similar profiles. In cases where the combination was common, participants were able to generate new clusters. Alternatively, if no matches were found, the note was labeled as an outlier and set aside for later discussion. More specific to the text itself, participants regularly engaged in search and scan tasks to find notes that contained certain words or phrases (e.g., 'try,' 'high-protein,' 'diet').

Clustering assistance: Because text was 'primary' for AD, and thus more salient for the participants, many of the initial clusters were based on text. However, participants consulted data attributes for working with these starting clusters. A commonly observed pattern was using data to *split* larger clusters into smaller ones. Specifically, participants used the cluster level visualizations to determine if the cluster could be split along attribute values (e.g., 'always cooks' vs. 'never cooks'). For a smaller number of instances, participants used data similarity for *combining* smaller clusters. Visualizations were also used to *detect outliers* in clusters and notes were moved or marked for further analysis.

Summarization assistance: Participants used data in a number of ways to *validate* their clusters. This included simple visualizations to test the 'purity' of clusters. Participants often hypothesized, and would test, the idea that people with similar themes to their quotes would share other similar properties. The data-derived similarity 'assessments' would often be captured as cluster labels. Participants also used data to develop a narrative across different clusters. For example, participants utilized their cluster summaries to find that *"...freshmen who live on campus and tend to eat unhealthily, then they become sophomores and juniors and start cooking, seniors live off campus... [but] this one group of seniors live on campus and do not eat healthy... they never moved on"*.

5.3 Design Guidelines

The probe sessions allowed us to identify key tasks for data assistance. These were used to drive many of Affinity Lens features. Additionally, we determined a set of guidelines both from observing the AD process and from feedback.

D1: Text first, then data. Affinity diagramming is at its most powerful when used for unstructured data, such as text. Datasets that are entirely structured are most often analyzed using other tools. AD, on the other hand, is suited to the bottom-up construction of clusters that requires human interpretation and input for clustering. Even in our probe, the two of five sessions that *began* clustering using data were less successful in completing tasks. They took a lot longer to analyze text within each cluster and to interpret how the text and data made sense as a whole. Because of this, Affinity Lens encourages end-users to start clusters based on analysis of text or other unstructured data. Though it would be relatively easy to implement, Affinity Lens does not, for example, suggest initial clusters.

D2: Support just-in-time insights. The type of data insights participants referred to during our study were highly context-driven and based on immediate decision support. Interactions to acquire such insights should be fast, expressive (support a variety of query and visualization needs), and low-effort, i.e., not distract from the primary task.

D3: Leverage spatial interactions for data access. Observing our participants we noticed extensive physicality to the AD process. Participants would move away and towards the wall to get different views. To understand the relationship between clusters (the broad view) they would often step away from the wall. To focus they would approach the wall and stand still (or seat themselves near the wall) to study individual clusters. A design guideline for Affinity Lens, and in part what motivated our use of AR through portable devices, was that the data could move with the AD practitioner and adapt to their spatial position and context. This is different, for example, from a large touchscreen that requires physical proximity for use.

D4: Offer automatic visual insights when possible. Though we encourage the text-first (D1) approach, this has the risk that practitioners over-focus and forget that other data is available. In our study, for example, we would occasionally 'probe' the participants to inquire if they required

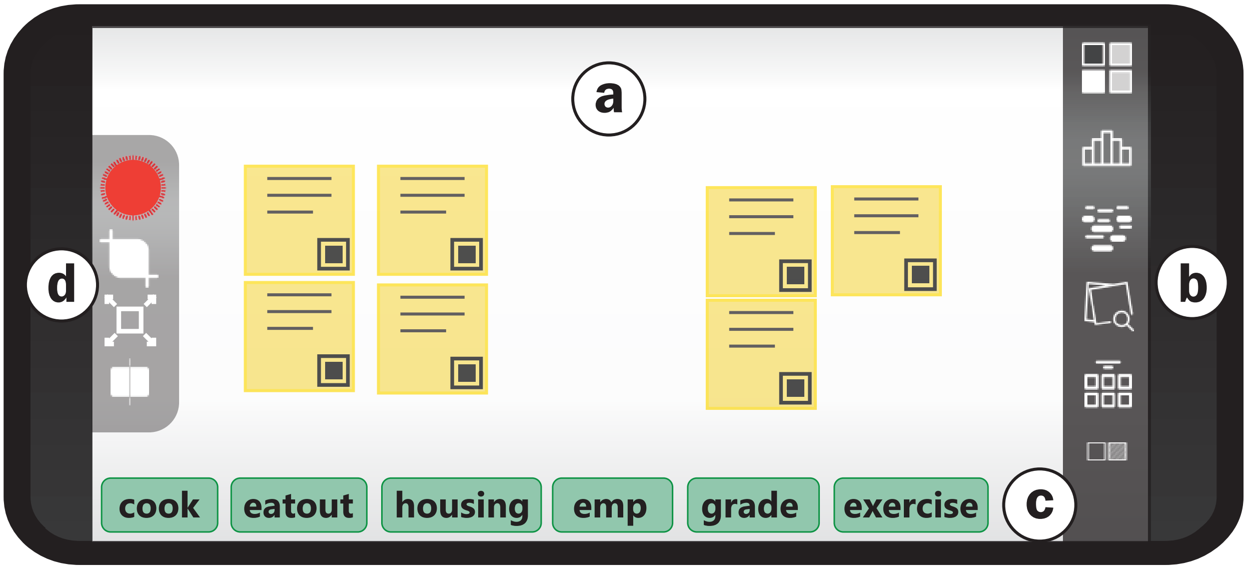


Figure 5.2: Affinity Lens User Interface. (a) main camera view, (b) contextual lens selector, (c) lens configuration options, (d) lens modes

visualizations. It was rare in our experience that participants would remember to initiate a data request, but were responsive when probed. When presented with the data, participants found the information helpful and in most cases performed actions based on the data. Affinity Lens must balance a 'background' role with active help. To achieve this, Affinity Lens is designed to keep track of the state of the AD process (as much as possible) and to be ready with a set of automatically generated visualizations when called upon.

5.4 User Experience

Affinity Lens was built as a mobile (phone and tablet) application, with a companion desktop utility for note creation and for viewing captured analyses. As opposed to an always-on display such as a projector or screen, mobile devices can be turned off when not needed (D1) and can be easily moved around in space to support near and far interactions (D4). Figure 6.3 captures the four main regions of the mobile interface: the largest, is dedicated to the camera and visualization augmentation (a), a contextual menu occupies the right edge of the display (b) and dynamically changes depending on what is present in the camera's field of view, a data attribute menu at the bottom edge manages the configuration of the current analysis tool (c), and dedicated controls allow setting modes of operation (d). In Affinity Lens, lenses are the collection of AR overlays available to the user. These include anything from visualization (e.g., bar charts based on what's in the display) to search (e.g., highlighting similar notes in the field of view). To better understand

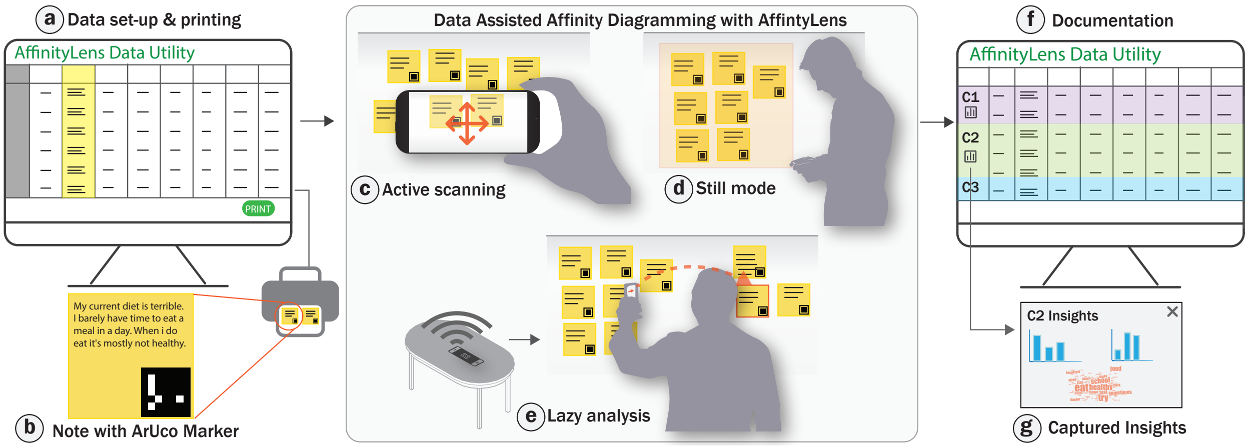

a Data set-up & printing
AffinityLens Data Utility
PRINT
My current diet is terrible. I barely have time to eat a meal in a day. When i do eat it's mostly not healthy.
b Note with ArUco Marker
Data Assisted Affinity Diagramming with AffintyLens
c Active scanning
d Still mode
e Lazy analysis
f Documentation
AffinityLens Data Utility
C1
C2
C3
C2 Insights
g Captured Insights

Figure 5.3: Affinity Lens workflow. Data is acquired (a) and automatically tagged for a Marker (b) for printing. Various forms of DAAD (c, d, e) can be documented (f) along with associated insights (g).

Affinity Lens' workflow (Figure 5.3) we follow a designer, Dave, who is working on a project about AI-based food recommendation for college students. Dave uses DAAD to analyze the food choice dataset to construct nuanced student personas to inform the AI design.

5.4.1 Data and Notes Set-Up

Dave begins his analysis by loading survey responses he's collected into our desktop utility application (Figure 5.3a). Each row corresponds to a different individual's response and each column is a question. From here, Dave selects the 'descriptive summary' column and issues a print command. Affinity Lens generates a unique AR marker for each row in the table which is printed along with the selected column value as affinity notes (Figure 5.3b). This 'binds' the printed note to the specific row. When using other data sources, such as interviews, Dave can import transcribed and coded notes from services such as nVivo, or even generate blank notes with markers and bind labels later using our lenses.

5.4.2 Clustering

Once the notes are printed, Dave lays them all out to begin the bottom-up clustering. He starts with a note that captures his attention: *"I try to eat healthy, but it doesn't always work out. . . "* He hypothesizes that this person may be unable to maintain a healthy diet, with planned, home-cooked meals, because they are busy. Dave picks up his phone with Affinity Lens, and *points* it at the note. Affinity Lens recognizes that only one note is in view, and augments the note using a lens that shows all attribute values (i.e., columns in the original CSV) associated with it (Figure 5.4 a). Here

Dave sees that the student *eats out* most of the time, and also *works* a part-time job. He taps on those attributes to mark them as important to that text. These attributes will be part of the resulting student persona. Further, Dave thinks that there may be other students with similar habits. He brings up the search lens and types in the keyword 'try' and then *pans* the phone over all notes (Figure 5.4 b). In the camera view of Affinity Lens, notes with the search term are highlighted in a different color. Dave gathers these notes as he finds them and piles them together for further clustering.

After forming a cluster of people which he labels 'tries but fails [to eat healthy],' Dave is interested in breaking it into smaller clusters. He brings up Affinity Lens and points it at the cluster. The view changes to offer a set of lenses that apply to note *clusters*. Dave is focused on this particular cluster, so he turns on the *still mode* (Figure 5.3 d) so he can continue working without pointing at the physical notes (D2, D3). Still mode captures a snapshot which persists in the display. He applies the heatmap lens by configuring different attributes, and sees that the cluster is split almost evenly by people who live on- and off-campus. Using this view Dave splits the cluster into two. Based on these clusters, Dave creates two data personas: one representative of students who live off-campus and work a part-time job, and the second who live on-campus. For each persona (and the associated cluster of affinity notes), Dave also has access to their food consumption data which is synthesizes using a word-cloud visualization at a later stage.

Next, Dave sets the phone aside and continues working on clustering. Affinity Lens continues analysis in the background (Figure 5.3 e) and alerts him that all but one student in the on-campus sub-cluster are first years (D4). By tapping on the notification, and pointing it at the notes (guided by Affinity Lens' navigation augmentation), he sees a heatmap augmentation in which one student is a senior. He marks the student as an outlier and places the note away from that cluster.

5.4.3 Pruning and Sensemaking

After clustering all notes, Dave sees that there are two clusters which are labeled "healthy eaters," and "healthy eaters + specific diet." He hypothesizes that those with a specific diet are more physically active. To validate this, he places both clusters in Affinity Lens' frame. From the lenses menu, he selects the histogram lens and configures it for the exercise attribute. Affinity Lens overlays individual histograms on top of each cluster, where he can see that those with specific diets tend to exercise more than the other group. He also looks at the distribution of health scores and finds that both groups have a similar distribution of self-reported health scores. To look for other text-based differences, Dave augments the two clusters with word cloud visualizations. He sees that the most used word in the healthy eaters is 'balanced,' while the other cluster includes words such as *high protein* and *paleo*. He saves these insights with their associated note cluster through

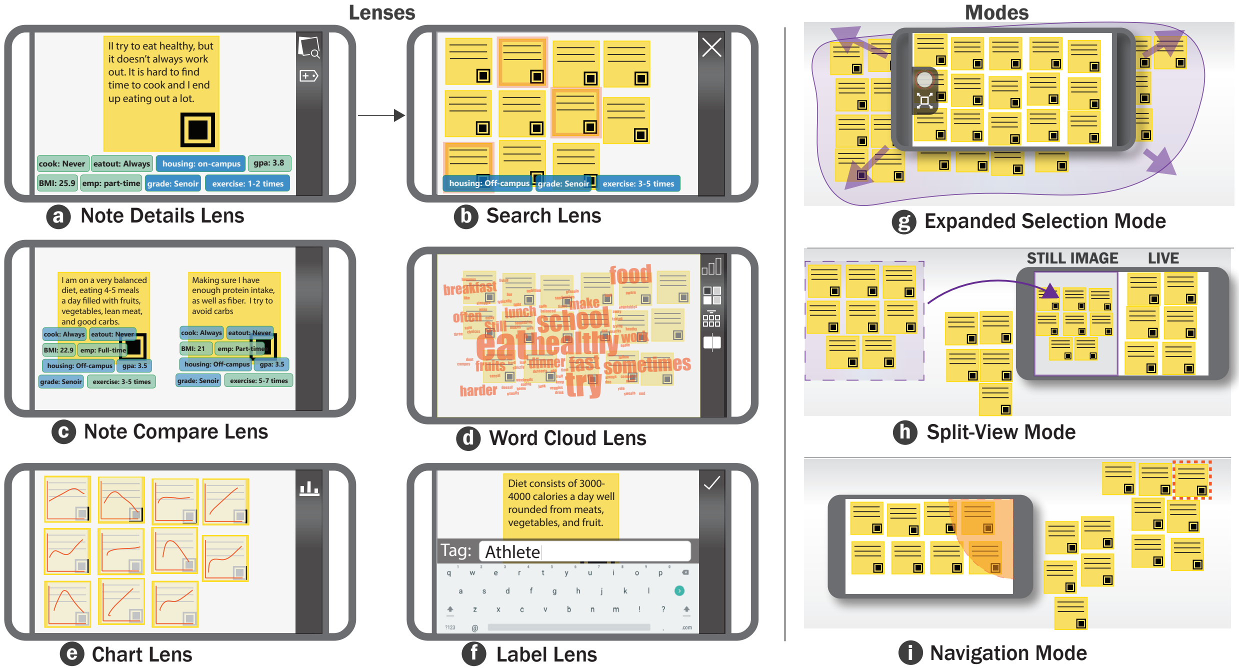


Figure 5.4: A sampling of Affinity Lens AR Lenses

the Affinity Lens interface. These visualizations and mixed-data cluster, inform two other personas including healthy eater with balanced diet, and healthy eaters with specific dietary requirements. Each cluster is associated with tag-cloud visualizations based on food-consumption data.

5.4.4 Documentation

Finally, Dave assigns labels to each clusters by using the label lens (Figure 5.4 f). Affinity Lens automatically updates the dataset with corresponding labels which can be viewed in real-time in the data utility tool (a web service viewable by Dave or others). At the end of the process, Dave has generated a set of nuanced personas that combines descriptive attributes about student diet with quantitative responses including living arrangement, employment, BMI, exercise, and summary visualization of food consumption. These data personas can support the design of the AI powered food recommendation experience.

5.5 Affinity Lens(es)

Affinity Lens allows users to choose among different lenses to overlay AR content on top of affinity notes. Here we describe the main categories and specific instances of lenses.

5.5.1 Lenses

For our prototype, we have implemented a number of lenses (examples in Figure 5.4) to support common tasks. These directly map to the four assistance types identified in our probe: details, search, clustering, and summarization. Affinity Lens is designed for extension so that new lenses can be added. In a practical scenario, users switch between different lenses as they engage in 'foraging' and sensemaking tasks.

Detail Lenses: In the context of mixed data, information contained on the physical note (i.e., the text) is only a *partial* view of the data. *Detail lenses* support understanding/interpreting the text by augmenting it with additional relevant information from the underlying data. In our implementation, when the end-user points at a single note, we augment that note with data values for that note (e.g., the row in the database). Other detail lenses, such as overlays of images [87] or videos, are possible with our architecture but not implemented in the prototype.

Search and Navigation Lenses: AD can have a large number of notes (as many as $200 - 500$ [99]). An advantage of using a digital aid such as Affinity Lens is that it allows users to find notes based on user-defined queries. We have implemented two search lenses that allow *searching by text* phrases, and *searching by data* attribute values. In our pilot study, we found that designers did not seem to want 'generalized' search queries. Rather they wanted to find 'similar' notes based on what they were doing. Put another way, they wanted 'search-by-example.' To support this, our *search lens* can be launched from notes viewed through a *detail lens* (D2). For example, when the designer points at the note, they see the associated data for that note through the detail lens. From this view, they can select *values* as search criteria (thus launching the search lens). Query results are displayed by the search lens by highlighting matching notes. The mobile device can be panned over the wall's surface and the lenses will automatically adjust the AR overlays to match the current view. Because not all matches may be in the field of view (D4), 'hints' are offered to indicate matching offscreen notes in the style of Halo [22] (Figure 5.4i).

Clustering Lenses: The Affinity Lens prototype supports grouping and clustering through three lenses: (1) the *heatmap lens*, (2) the *note comparison lens*, and (3) the *cluster label lens*. The *heatmap lens* places an overlay on notes that uses color to encode a selected attribute and its values (Figure 5.1a). For example, we might select 'weight' as an attribute and all notes will be color coded from light to dark based on the weight value associated with that note. This form of augmentation acts to summarize but also directly supports decisions around splitting and grouping multiple clusters. For a pair of notes, the *note comparison lens* (Figure 5.4c) displays those data values that are the same and those that are different (a weak representation of affinity). Finally, the *cluster label lens* is used to 'tag' all notes in a cluster with a persistent label.

Summarization Lenses: The final set of lenses allow end-users to summarize insights about clusters. This is done largely through the use of visualization overlays. In addition to the heatmap lens, our prototype also provides a *histogram lens*, a *wordcloud lens*, and a *radar plot lens*. The histogram lens will generate a histogram bar chart based on some selected attribute (e.g., the number or fraction of people who said 'yes' to dieting in a cluster versus 'no'). Clusters can be explicit (i.e., the designer tagged a cluster) or can be dynamic and contextual based on the notes in the field of view. The resulting histogram is placed over the entire field of view. When looking at text, a wordcloud lens (Figure 5.4d) will generate an overlay of common words (sized by frequency) on top of the notes. A radar lens will produce a radar plot to summarize multiple quantitative variables simultaneously. When multiple clusters are in view, or the designer uses a split view to see two clusters side by side, summarization lenses will be applied to each cluster separately (e.g., two or more histograms will be overlayed).

5.5.2 Interactive Querying through Scene Specification

In Affinity Lens, the primary mode of interaction is by first selecting the lens (and potential parameters on the mobile device's screen) and then viewing the physical notes through the phone's display. The subset of notes in the view provides a natural scope for the query (D3). The user can either use Affinity Lens in *live mode*, where the display updates based on the camera's field of view, or in *still mode* which uses a static snapshot. In live mode lenses dynamically adapt as the user pans across the surface. In still mode, the user can easily switch between multiple lenses and apply them to the notes captured in the view. This can be significantly more comfortable than continuously holding the phone in mid-air and also allows for 'private' analysis in a shared setting. To support analysis of large clusters, we provide an expanded selection mode. The mode will cause Affinity Lens to include off-screen notes, that were labeled as belonging to the cluster, in any analysis (e.g., a histogram) (Figure 5.4g).

In either *live* or *still* mode, the user has the option to 'split' the view (Figure 5.4h). This permits comparison between different clusters that are physically distant. It also allows for an overview-plus-context view where one side of the screen can be used to drill down into details for notes or clusters contained on the other side of the screen.

Finally, Affinity Lens supports what we call *lazy interactions*. Affinity Lens leverages periods of inactivity to analyze data and generate potential clusters and other insight recommendations such as outliers. When a new insight is available, the phone displays a notification to the user about the insight along with details about the target set of notes. The user can then tap on the insight and use guided navigation to find the physical notes on the wall. For example, if Affinity Lens detects an outlier in a particular cluster when the notification is selected, arrows will lead the user in *live mode* first to the cluster and then to the highlighted outlier.

5.6 System Architecture

While complete details of our implementation are beyond the scope of this paper, we provide a high-level view of the architecture. As shown in Figure 5.5, Affinity Lens is comprised of five main components: (1) Scene Analyzer, (2) Lens Controller, (3) Dynamic View Configurator, (4) lenses, and (5) the Data Access and Analytics Module.

The *Scene Analyzer* detects notes from the incoming camera feed (i.e., the scene) by using computer vision based processing. Note information including the number of notes and positions are relayed to the *Lens Controller*. This module determines candidate lenses based on notes and updates the phone interface through the *Dynamic View Configurator*. Once a lens is selected and applied (either the system default or by end-user selection), the system generates a database query for the notes in view for execution by the *Analytics Module*. Finally, query results are rendered on top of the scene by the View Configurator. This process happens continuously and in-sync with the camera feed. The system itself is implemented using JavaScript and is executed (and displayed) in the browser on the phone or tablet device.

5.6.1 Scene Analyzer

Our current prototype uses ArUco Markers [86] for detecting notes along the x-y plane. Using computer vision libraries [54, 173], this module determines marker positions and builds spatial relationships between notes. The scene analyzer overlays a grid structure on top of the markers, and each marker is assigned a row and column position relative to the scene. This information is also used to detect clusters in which individual clusters are separated by areas of empty grid cells. In each refresh cycle of the scene, notes are updated with revised x and y positions along with marker IDs for eight adjacent markers (to support navigation), and cluster ID. This information is used by other modules in the system pipeline.

5.6.2 Lens Controller

This module consists of a collection of lenses, along with a look-up table containing prerequisites and configuration parameters. Depending on the number of notes or clusters in the scene (single, pair, multiple, etc.), the lens controller will select all applicable lenses and send configuration information to the Dynamic View Controller. If the mode corresponds to a single lens, the controller also instantiates the detail lens. This module also coordinates different lenses by passing relevant setting and parameters between them (e.g., maintaining attribute selection between lenses, setting selected attribute values such as search parameters, etc.).

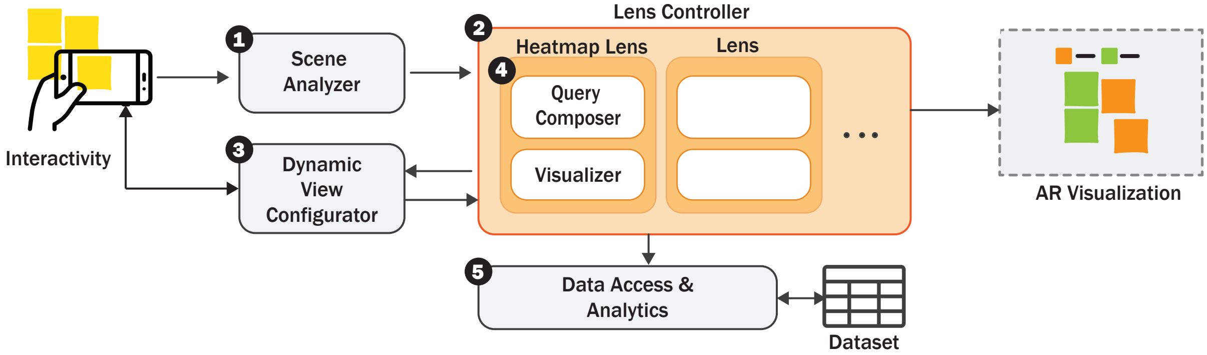


Figure 5.5: System Architecture. (1) Scene analyzer extracts notes from camera feed, (2) lens controller determines set of lenses applicable to notes in view, (3) dynamic view configurator updates the interface with available lenses, (4) lens queries for data from the (5) Data access and analytics module, and renders the augmented visualization.

5.6.3 Dynamic View Configurator

The Configurator updates the Affinity Lens interface in real time based on input from the lens controller. Candidate lenses are presented as items on the right contextual menu. When a lens is selected, appropriate configuration attributes are rendered at the bottom of the screen. When the end-user interacts with these menu options, this module also relays events and selections back to the lens controller. Once a lens is selected, this module applies the output of the lens and displays the augmented view on the screen.

5.6.4 Lens Design

Each lens is made up of two sub-components: a *query-builder* and the *visualizer*. The query builder constructs a query for the notes in the view along with other lens specific configurations (e.g., selected attribute). For example, the histogram lens will identify that a cluster of notes is currently in view and query the database for the values for *those notes* based on the attribute the end-user has selected. This query is processed by the Data Access Module. For example, when a histogram is requested over a set of ten notes, with ‘living preference’ as the data attribute, the query builder fires a query by passing note IDs and living preference as conditional clauses. The results are rendered by the visualizer sub-component. This module makes use of positioning information made available by the scene analyzer to determine the placement of the rendered visualization. This abstraction allows us to easily build new lenses through a standard API.

5.6.5 Data Access and Analytics

This module supports two types of data operations. It executes query requests issued by the lenses over the dataset and updates the dataset based on real-world actions (e.g., if a new cluster is formed and detected, the associated rows in the database are labeled with a cluster ID).

The module also supports lazy-analysis interaction. Based on note adjacency and clustering information provided by the Scene Analyzer, background clustering and analysis are executed and results are surfaced back to various lenses. For example, to support clustering, we use the techniques developed in the iCluster work [21]. Existing cluster information is used to create a metric space by which clusters are formed. Distances between notes are based on a combination of attributes and keywords. Weights on attributes are adjusted such that notes belonging to the same cluster are deemed closer together while notes in different clusters are further apart. If there are sufficient notes in each cluster, a classifier can be trained to help decide which cluster a note belongs to. Using this information, possible clusters can be highlighted for a selected note. Alternatively, if a cluster is selected, matching unclustered notes can be highlighted.

5.6.6 Implementation

We implemented Affinity Lens as a mobile web application that runs on the phone browser. A Node.js server handles data analytics and image storage, and a HTML/JavaScript client uses OpenCV.js and js-ArUco libraries for camera and image processing and D3.js for visualization.

5.7 Evaluation

To evaluate Affinity Lens, we conducted two different in-lab AD studies. The first was a controlled study (using the same protocol as in section 6.2) in which we determined whether end-users could effectively generate data insights using Affinity Lens. In the second study, which was open-ended, we aimed to evaluate Affinity Lens in a realistic AD workflow.

5.7.1 Study 1: Controlled Evaluation

For this study, we conducted three 90-minute sessions (two participants per session) with four HCI design student (P1-P4) and two UX professionals (P5-P6). We used the same task and study protocol as in section 6.2, but instead of having the data directly printed on the notes, we added an ArUco marker to bind the note to a data row. To encourage discussion between participants (for think-aloud), we only provided a single Android mobile device (5.5. inches,1440 x 2560 pixels) with Affinity Lens running on the Chrome browser.

At the start of the session, participants were given a hands-on demo of the system including the use of different lenses. Once participants indicated familiarity with the system, they proceeded to analyze and cluster the notes for the advertising task. Sessions were video recorded for analysis, and a study coordinator took observational notes. At the end of the session, participants did a verbal walk-through of the distinct characteristics of each cluster and finally took part in an informal interview to report their experience.

Findings

Data assistance for clustering notes: Across all sessions, we observed that participants effectively invoked different lenses to generate data overlays for single, and group of notes (D2). While reading a note, if participants noticed an interesting phrase, or when there was disagreement about which cluster to place the note in, they would invoke the details overlay on that note. Beyond note level details, participants also made use of data overlays to revise initial clusters generated from text. A repeated pattern we observed was that participants cycled through different data attributes using the heatmap lens to split a clusters, *update* cluster labels, or make distinctions across different clusters.

A common practice in AD is to set aside notes that do not fit into any clusters for further analysis. For such notes, participants took a trial-and-error approach by placing the note being discussed next to notes in other clusters to test for "affinity" using the note-compare overlay. Once clusters were generated, participants used both the histogram and heatmap overlays for validating cluster affinity and outlier detection (D4). They often expressed delight when their text-based interpretations matched what the data showed. However, participants reported that they did not find the wordcloud lens very useful. We suspect this is because of the smaller number of notes used in this study. Further, we only observed a few instances of multiple-cluster comparison. This may be attributed to the fact that data level bins were already determined when clustering.

In all sessions, while the clusters aligned with our artificial grouping, we observed that overall engagement with Affinity Lens was *higher* than we had intended (i.e., somewhat a violation of D1). This may be due to the nature of the clustering task which required data insights, but more likely the novelty of the system. As reported by P2: *"I was relying too much on the app . . . not using the notes as much"*, and P1:*"it (system) is fun . . . especially when you don't know how to group something (using text)"*.

User Experience with Affinity Lens: The portable nature of our solution made it easy to blend spatial interactions with our lenses interface (D3). In one of the sessions (P1-P2), participants spread the notes on the table, and sorted the notes by using the heatmap lens. When discussing cluster level insights, participants found the *still-mode* extremely useful. We observed that one of the participants would capture cluster insights and engage in rich discussion with the other participant by trying out different lenses (D3). Participants also found the *split-view* mode helpful when comparing distant clusters, and appreciated that they did not have to move clusters around to make comparisons.

During the feedback session, all participants reported that the concept of lenses, and Affinity Lens' interface was easy to understand and use. When explicitly asked about the ArUco markers, participants indicated familiarity with QR codes, and that the markers did not interfere with AD. We note that in some instances, Affinity Lens did not recognize the markers. For example, distance was an issue when multiple clusters were in view. This issue can likely be remedied by implementing image enhancement techniques (e.g., [221]).

Finally, in comparison to our probe session, in which data persisted on notes along with text, the AR features of Affinity Lens made it possible to make salient (bring to view) specific types of details, on demand. Participants were able to easily toggle between text and data views, and compare insights across clusters in a fast and fluid manner. A drawback is that data insights are not persistent, which can be problematic when working with larger datasets. As mentioned by one participant (P5), persisting data-specific insights on paper might be useful. They even recommended having colored markers corresponding to the heatmap color palette, and adding indicators on physical notes (they referred to clusters by colors: "these are the reds, add them to the purple cluster").

5.7.2 Study 2: Open-ended AD Workflow Evaluation

To emulate a realistic workflow as described in section 5.4, we gave participants the results of a survey we conducted about Facebook Usage and Privacy using Qualtrics. The survey consisted of closed-ended questions about Facebook Usage, Ads on Facebook, types of data shared (posts, pictures, profile information, etc.), concerns about privacy and data sharing, and an open-ended question requesting examples of privacy violation on Facebook. All questions were required, and we set a minimum limit of 250 characters for the open-ended question. We collected 100 responses using Amazon's Mechanical Turk and generated the notes by exporting the data as a text (CSV) file from Qualtrics.

We recruited six participants with prior experience in conducting AD: three UX professionals (P7-P9), one design-science researcher (P10), and two privacy researchers (P11-P12). We conducted three sessions with pairs of participants, and each session lasted 2-hours. Participants were paid $30 for their time. In each session, we described the survey questions to the participants and asked them to generate sources for privacy violation using AD. We then provided a guided tutorial of the system. We concluded each session with a walkthrough of the clusters and an informal interview. In this study, we provided participants with multiple device options (phone, and tablets with 12.3-inch screen, 2736 x 1824 pixels) all running Affinity Lens on the Chrome browser.

Findings

Data-assisted, not data-driven clustering: In all our sessions, we observed participants trying to navigate when to use data versus text views. At the start of each session, one of the participants wanted to start with the data view, while the other preferred generating an initial set of clusters based on text (P11: *"open-ended responses are more reliable . . . we can use our judgment to categorize them first and then use [Affinity Lens] to double check"*). The rationale for data-first was that being able to quickly try out different groupings with data would help ask more questions earlier on in the process, as P9 mentioned *"rather than using the lenses to drill-down, I wanted to use it as a way to bubble-up questions."*

While data overlays offered a quicker alternative to generate clusters (P7: *"we started with the obvious and it was massive. . . we realized we need to get out of reading the content and look at the data"*, P8: *". . . with all the ad tracking we wanted to hack for a trend,"*), participants realized that over-reliance on data could make it hard to make sense of text content within individual clusters. The switch from data-view back to content occurred when participants became aware that they devalued content, or when there were no discernible patterns from data. In summary, participants saw value in having both views, and being able to switch between them (e.g., P11: *"[Affinity Lens] enhanced the depth of analysis and helped us figure out what is going on, the nuances. . . "*).

Time costs for DAAD: When using DAAD, we hypothesized that Affinity Lens would speed up the AD process. Across all sessions, we observed variations in when, and for how long, participants engaged with Affinity Lens. In session 1, the use of Affinity Lens (i.e., data view) was more evenly spaced out. The first use was at 14.5 minutes into the session, followed by switching between text and data views every 10-12 minutes. In sessions 2 and 3, participants first used Affinity Lens after around 40 minutes of clustering by note content but extensively used Affinity Lens for pruning and sensemaking during the second half of the session.

Some participants felt that they spent *more* time on AD because the insights from data were interesting (e.g., P7: *"If I had just words I would have been like, yeah, that is all we are going to get . . . [with Affinity Lens] I could keep going on looking for new patterns"*). In this study, because participants were not directly involved in survey design, some participants found the range of attributes overwhelming (we utilized a very broad survey instrument). P8 suggested different tabs to categorize the attributes (e.g., demographics tab, Facebook usage tab, etc.) but added that if they were using in their own work, this may not be a problem.

DAAD in existing design workflows: In discussing applicability of DAAD in their own design process, several participants were keen on using Affinity Lens as a way of getting "buy-in" from managers and data analysts. For example P7:*"not everybody buys into AD and Affinity Lens is a nice vis bank . . . "*, P9: *"I could advocate for the realness of my work. . . "*, etc. While all participants agreed that quantitative data was not the place to start AD clustering (confirming D1), participants mentioned that data insights from AD could generate an initial set of hypothesis for data analysts. During feedback, participants also recounted examples from their own experiences of working with mixed methods approaches, and how Affinity Lens could have helped in those situations. For example, P4 mentioned conducting AD exercise with data collected from a photo diary, and that having Affinity Lens could have helped augment pre- and post-study information and metadata (e.g., timestamp).

In summary, the results from our study demonstrate the usefulness of Affinity Lens in the AD workflow. Though we expect that testing Affinity Lens in additional contexts will lead to more features and improvements, the feedback we received from our participants, and their interactions with Affinity Lens, is highly encouraging.

5.8 Discussion

There is clearly a need for integrated sensemaking from qualitative and quantitative data when conducting mixed-methods research. Through Affinity Lens's AR overlays, we demonstrated how DAAD can enrich the analysis experience of survey data, a typical use-case within HCI research. Beyond surveys, HCI work also uses interaction logs, sensor streams, and multimedia content (photos/videos) to inform system design and end-user behavior. Current workflows for analyzing such data typically follow a unidirectional pipeline (e.g., video footage –¿ transcripts –¿ grounded theory coding), making it hard to flexibly combine raw data with qualitative insights in a just-in-time manner. Future work can look at ways to incorporate DAAD into existing workflows by linking lenses with rich data sources (e.g., [158]). For example, one can augment the text from think-aloud transcripts with interaction logs showing mouse clicks data, or overlay raw video footage of actual task execution for multiple participants (affinity notes) in parallel.

In our current implementation of DAAD, we do not focus on the collaborative nature of AD, or potential collaboration between qualitative and quantitative analysts. However, we believe there is an opportunity for more collaboration-focused lenses. For example, we can imagine sharing configured lenses between devices to enable different users to study different parts of the wall with the same lens. Further, in Affinity Lens we primarily support just-in-time insights with minimal query specification (D2). To emphasize the assistive role of data, and given the form factor, we did not explore all features of a data analytics tool such as Tableau or R in DAAD. However, based on participant feedback it may be desirable to have richer support for data analysis within DAAD to enable collaboration between designers and analysts. Building on prior work on spatial [11], and tangible visualizations [83, 132], we are exploring ways to leverage sticky-notes for querying and visualization specification.

In our studies, we printed notes on plain paper. This requires the added effort of cutting and adding adhesive. In real world deployment, this limitation can be easily overcome by either using a template based printing technique (i.e., pasting sticky notes on letter size paper template before printing) or by using special portable printers such as [50]. Lastly, camera resolution and field-of-view (FoV) constrain scalability when there are a large number of notes. This creates a challenge for using the phone for maintaining the system's internal models of the physical AD. Affinity Lens currently updates note positions by surreptitiously capturing frames when the user pans the phone during use. Future work can explore other active interactions to maintain this representation (e.g., prompting the end-user to explicitly capture "current state" by scanning across the wall). By open sourcing our implementation, we hope that we can better understand how these features are used and enhanced.

5.9 Summary

By combining qualitative and quantitative views of end-user data, Data Personas serve as a useful probe in designing AI-powered user experiences. While Affinity diagrams lend themselves well to creating personas from raw qualitative data, traditional solutions don't readily support generating data personas. Furthermore, in working with mixed data sources, designers require analytical power beyond physical sticky notes. Prior research to address these shortcomings has posed barriers, including prohibitive costs of large, interactive whiteboard systems or disruptions of current workflow practices. With Affinity Lens, we have demonstrated how data-assisted affinity diagrams can be implemented with low-cost mobile devices while maintaining the lightweight benefits of existing AD practice. To date, we have only lightly explored the space of lenses, but already, users of the current system were enthusiastic about using Affinity Lens in their current AD-related work tasks. As discussed in Chapter 3, using DAAD, designers can work with AI engineers to construct nuanced user-segmentation to inform AI's training data requirements. Further, these user segments and data persona will allow teams to brainstorm about AI's feature and behavior design and instantiate AIX prototypes using distinct data personas. The next chapter investigates how UX designers can incorporate data personas and machine learning models into AIX prototyping workflows.

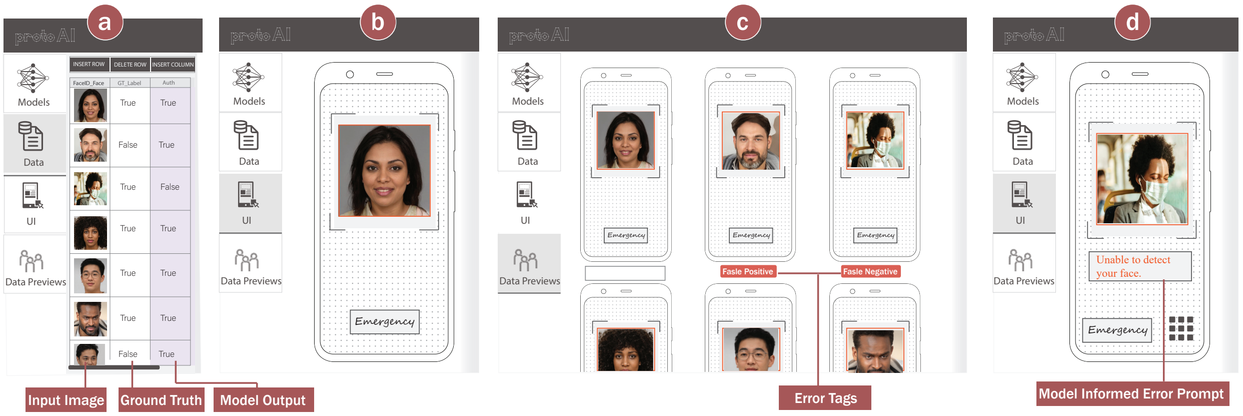
CHAPTER 6

Model Informed Prototyping

When prototyping potential designs for user interfaces (UI), designers work to transform end-user needs into interface specifications [246]. By taking a *top-down* approach, designers: (1) express user requirements as task-flows; (2) map task items into graphical user interface (GUI) objects; and (3) assess different task-to-GUI mappings against end-user needs to finalize the design [162, 246]. For instance, to design a phone 'unlock' user experience (UX), the designer may consider interface alternatives—such as an alpha-numeric password, a numeric passcode, or pattern-based unlocking—to allow end-users to input identifying information for access. By assessing those alternatives against user needs (e.g., fast to unlock, secure, low cognitive effort to remember), the designer will finalize the UI design. However, when prototyping AI-powered applications, such a top-down approach is impractical [253].

AI-powered applications bring additional challenges to UI prototyping. AI features introduce dynamic behavior due to the scope of training data, system use over time, and variations in input data individual users contribute and the potential to learn from outcomes. Thus, designers must identify the *interactions* between user task-flows and AI capabilities [46, 62, 117] in order to design the user interface for AI experiences (AIX). By exploring AI's capabilities and limitations through prototyping, they need to design interface adaptations such as explanations for AI outputs, seamless handling of AI failures, and collecting user feedback to improve the AI [10]. In the process, AIX designers also need to assess interface choices against diverse users and contexts of use.

Unfortunately, current UI prototyping tools lack support for designing AI-powered interfaces [250]. By assuming a 'black-box' view of AI, tools make it challenging for designers to access necessary AI attributes during the design process [235]. Prototyping tools also lack support for iterative testing of AI features through a "fail fast, fail often [251]" approach. For a *AI-powered* phone access using face identification (ID), current tools can at best show where to display the camera field of view on the interface and design static error messages. However, without exploring the AI's behavior first-hand, the designer may not know what inputs the AI needs (e.g., head frontal-view). They may fail to understand how accurately the AI can perform, when it might fail,



FaceID_Face	GT_Label	Auth
	True	True
	False	True
	True	False
	True	True
	True	True
	True	True
	False	True

Figure 6.1: Prototyping a Face-ID Phone Unlock user interface. The designer (a) provides a set of portrait photos and runs the Face-ID model, (b) prototypes the interface using a design-by-instance approach; (c) ProtoAI generates instances of the interface for all of the photos and flags false positives and false negatives. The designer (d) updates the design to include a numeric keypad and shows error message based on AI model output.

and how to prompt users experiencing failure (e.g., by asking them to move closer to the camera). To prototype AI features, designers currently need to work with multiple tools to explore AI behavior (e.g., [176]), probe its capabilities and limitations [93], and evaluate their design with diverse user inputs (e.g., skin-tone, lighting conditions, camera angle, facial features such as beard, glasses, or a mask) [40]. This introduces friction to the rapid prototyping process [24, 213]. Thus, the motivating question for our work is: *How might prototyping tools allow designers to directly incorporate target AI features during rapid and iterative prototyping?*

Through an analysis of current human-AI (HAI) design guidelines from academic and industry sources [10, 92, 128], we identified a set of needs and design considerations for AI prototyping tools. To maximize end-user success with AI features, designers need to optimize UI design through *vertical* end-to-end prototyping [24]. They also need to identify different kinds of interface *breakdowns* such as mismatch with end-user expectations, low utility (high cost) from using AI, and data specific failures and offer repairs to recover the user experience. Collectively these tasks require that designers can simulate their interface designs with different data and model outputs. To accomplish this, we propose *Model-Informed Prototyping* (MIP), a workflow that combines model exploration and interface design tasks.

In our system implementing MIP, *ProtoAI*, designers can directly run target machine learning models by providing input data and then incorporate the model's outputs in their UI prototypes. Instead of placeholder content, ProtoAI's *design-by-instance* approach allows designers to experience the AI's behavior first-hand as they are designing. Further, ProtoAI automatically generates data previews of the UI for differing input data, allowing designers to evaluate designs for breakdowns across diverse scenarios and contexts of use. This enables them to decide how best to integrate AI features into end-user's tasks and offer necessary adaptations for AI's uncertainties. As shown in Figure 6.1a, to design a Face-ID phone unlock AIX, the designer can begin with a diverse set of registered and new faces along with ground truth data as inputs to the *Face Identification* model. After running the model, the designer can prototype the Face-ID user interface using one of the input faces and corresponding model outputs (Figure 6.1b). In the data previews tab, ProtoAI automatically generates previews of the interface for each input data, allowing the designer to evaluate the designed AIX for diverse inputs (Figure 6.1c). ProtoAI automatically tags errors such as false positives and false negatives based on ground truth data. By analyzing errors, the designer can revise the interface design by providing alternative login options and displaying data specific prompts to recover from errors (Figure 6.1d).

MIP streamlines model exploration and UX design tasks during the prototyping process for AI-powered interfaces. By extending the familiar design paradigm of current prototyping tools, ProtoAI allows designers to operationalize HAI design guidelines within their created designs. Based on feedback from designers, ProtoAI lowers the barrier to data-driven design required in prototyping AI features. Our key contributions include: (1) *Model-Informed Prototyping* – a new workflow for prototyping AI-Powered applications, (2) ProtoAI, a tool that implements MIP for GUI, (3) results demonstrating how our approach can support different types of AI breakdowns and repair.

6.1 Related Work

The user interface design process consists of a series of transformations between end-user task requirements and the user interface syntax [246]. Standard UI prototyping tools such as Wireframe.cc [245], Figma [80], and Adobe XD [1] allow designers to work at the user interface level alone through *horizontal* prototyping [24]. However, when designing AI-powered applications, both the end-user task requirements and the underlying AI components needs to be mapped onto the user interface syntax [43, 46, 92]. This requires a form of *vertical* prototyping in which designers can access specifications about the underlying AI implementation and map them to AI-powered interfaces [24, 242]. In ProtoAI, our goal is to address this need by designing a vertical prototyping tool for AI-powered interfaces. A recommended workflow for UI prototyping consists of three phases: design, test, and analysis. A number of UI prototyping tools (including our own) follow this model [104, 143, 156]. Here, we describe requirements and techniques from prior literature for each phase as applied to AI-powered interfaces.

6.1.1 Design

Numerous guidelines exist to design AIX by considering the intersection of human-centered needs and AI capabilities [10, 90, 92, 107, 120, 128, 239]. However, to *operationalize* these guidelines, designers need access to the AI model in order to map its characteristics to the UI syntax [62] (see Section 6.2). For instance, in mixed-initiative design, AI systems automatically act on end-users' goals (when clear) and use interface 'dialog' to resolve any uncertainty [120]. However, the specific dialog in the UI depends on the underlying AI and input data-context. In this regard, prior work has looked at using data as a material for AI design [146, 111]. Just as engineers prototype ML models, designers can begin with 'minimum-viable-data' and iteratively incorporate additional data for diverse users and contexts [166, 235]. This allows prototyping of AI interfaces from the inside-out: from the data model to UI [3]. Mixed-fidelity prototypes [172] could allow designers to incorporate high-fidelity data elements in early-stage prototypes to represent ML's dependency on data [68]. In ProtoAI we take a similar approach and allow designers to incorporate input data and ML model outputs into UI prototypes (e.g., designing password meters by mapping scores from neural networks and heuristics to a visual bar [233]). Further, given most designers' limited expertise with AI [252], prototyping tools should make AI features more accessible, immediate (support rapid iterative feedback, reflection-in-action, and reflection-on-action), and generative (allow test, probe, and exploration iterations) for UI designers [45, 98, 154].

6.1.2 Test

AIX designers need to map AI-to-interface features, identify gulfs of execution and evaluation, and assess visual aesthetics for AI features. Further, they should evaluate whether their design is robust to AI's unpredictability [117]: How does the AI-infused interface react to a diverse set of data and contexts of use [43, 235]? Building on existing UX practice, designers may consider approaches such as constructing personas with varying quantitative data [195]. Wizard of Oz (WoZ) testing is also effective for evaluating early-stage prototypes [34, 69, 171], and a number of data-dependent systems implement digitally scaffolded 'wizards' for testing prototypes during design [59, 104, 143, 156]. For instance, Suede implements electronically supported WoZ testing techniques that generate chat messages using test data [143]. In Topiary [156], designers create a map that models people's location, which the Wizard uses to update locations during testing. In ProtoAI we automatically generate interface alternatives by invoking built-in models with input data provided by designers. This lets designers experience the UI's design first hand [37]. In addition, conventional interface design methods include indicating how the UI should behave through demonstration by examples (e.g., [181]). Inspired by this approach, in ProtoAI, we allow designers to configure desired behavior (ground truth) by providing model output data for comparison (i.e., designer as wizard [34]).

6.1.3 Analyze

To analyze performance at the AI model level, engineers use summary statistics such as accuracy, precision, and recall. Tools exist for engineers to analyze the overall performance and look at individual data points to reason about model failures (e.g., [9]). Designers need similar analysis and visualization tools at the interface level that will allow them to identify mismatches in model behavior. For instance, D.tools offers a 'group analysis' mode aggregating data from multiple user sessions into one view [104]. The What-if tool [93] allows designers to see the confusion matrix for binary classifiers visually [93]. Designers should also be able to incorporate subjective metrics at the intersection of model performance and UX (e.g., subjective perception of errors [144]). In ProtoAI we support subjective analysis through designer generated tags and visual summaries. During iterative prototyping, the goal is to identify breakdowns in design and offer fixes [27, 91, 96, 200, 244]. For instance, through iterative UI experimentation, Quick Access identified UI needs to offer proactive recommendations [231]. The DECOR system characterizes multi-device responsive UIs as a design repair problem and offers techniques for efficient repairs [214]. ProtoAI's instantiation of UI for different data points allows designers to analyze AI-feature breakdowns without performing mental simulations of differing data contexts. Moreover, the generated previews provide the necessary context to make effective repairs [106].

6.2 Design Considerations

A primary objective when prototyping AIX is to maximize end-users' success. In this regard, both academic and industry sources have put forth design guidelines about good AIX design [10, 92, 128]. With ProtoAI, we want to make it easier for designers to operationalize these guidelines in their interface designs. We collected a total of 284 Human-AI design guidelines. We conducted inductive *in-vivo* coding to synthesize the main objectives and tasks for designers and the corresponding AI components necessary to accomplish those tasks. We find that the guidelines offer best-practice recommendations to map AI features into UI design patterns (and end-user tasks). This includes making decisions about automation, AI assistance, and human-effort by aligning AI capabilities and end-user needs. More importantly, the guidelines prescribe design 'fixes' to lower end-user impact from AI-breakdowns such as (1) *end-user context breakdown*: AI performs poorly for some user-data and in some usage contexts; (2) *expectation breakdown*: AI behavior and outputs do not align with end-user mental models; and (3) *task-utility breakdown*: higher cost of using AI due to its failure to understand end-user goals. To address these break-

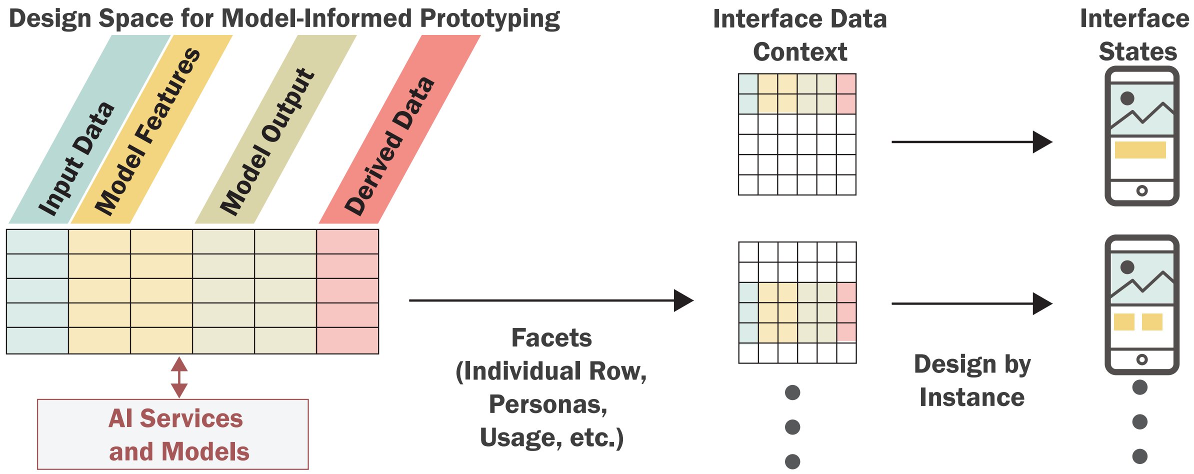


Figure 6.2: Design space and workflow for Model Informed Prototyping.

downs, designers need access to the underlying AI model, features, and output data for diverse end-user inputs. On this basis, we derived a set of design considerations for AIX prototyping tools (i.e., model-informed prototyping).

D1: Prototyping tools should allow designers to invoke ML models by specifying input data directly. When prototyping AI features, designers need to choose whether to automate the task entirely, ways to augment human effort with AI, and whether the AI should be proactive or reactive (acting only upon human invocation), etc. [128]. The objective is to minimize interference with end-user tasks while maximizing utility [120]. To make these choices, designers need to understand how the AI performs for different input data, what output it returns, and under what conditions it might fail. This will allow designers to incorporate AI features into end-user tasks appropriately. Further, designers need access to AI model features to offer the rationale behind specific outputs for specific users. For instance, they may want to present confidence in the model's output or show "why" messages to end-users to design for transparency and trust. To realize these design objectives, designers need to evaluate model performance for potential end-user inputs during their prototyping process.

D2: Prototyping tools should allow designers to incorporate AI outputs into interface design. When designing AI-powered interfaces, guidelines recommend that the AI outputs and UI presentation be aligned to avoid cognitive distortion. Further, the AI-generated content should be visually different to allow end-users to adjust their expectations about AI features (and, in turn, diminish frustration). In mixed-initiative design, designers need to find the right presentation based on confidence thresholds (e.g., showing only the high accuracy item, ranking items as a list, etc.). Designing for these guidelines could benefit from instance-based prototyping by directly incorporating model outputs into interface design. This will give designers a more accurate representation than placeholder elements when making design choices (e.g., presentation layout, conditional logic for UI, error presentation, guided-recovery from failure, feedback controls, etc.).

D3: Prototyping tools should allow designers to shape model APIs according to end-user needs. Based on decisions about AI feature integration into interface design, designers may need to revisit the model inputs and outputs (i.e., the API). Design guidelines recommend that AI model APIs be designed based on principles of information architecture for interface design. For instance, designers may need to split complex outputs and explanations into multiple parts and present them one at a time. When presenting statistical or numeric outputs, the designer needs to consider factors such as precision and rounding. In cases when numeric values are not appropriate, designers should determine appropriate mappings to categorical variables. This is particularly useful when presenting recommendations along with explanations [92]. Prototyping tools should allow designers to flexibly transform model output and feature values into end-user-friendly formats.

D4: Prototyping tools should allow designers to evaluate design choices across diverse users and usage contexts. With conventional applications, design typically *converges* to a set of standard features across all users. However, with AI models, we can personalize the end-user experience to highly specific contexts. With this intent, HAI guidelines recommend applications should be designed to work across a diverse set of users, use cases, and contexts of use. For example, "while all errors are equal to an ML system, not all errors are equal to all people. [92]" To operationalize these recommendations, designers need to evaluate their interface choices across diverse data. Prototyping tools should allow such evaluation to test and analyze the impact of unwanted model behavior that could negatively impact users. Tools should also support evaluating how AIX could evolve over time and how the interface should adapt accordingly.

D5: Prototyping tools should allow flexibility for designers to incorporate model-related data rapidly and iteratively. Based on our analysis of the design guidelines, we formulate a design space for Model-Informed Prototyping. As shown in Figure 6.2, MIP's design space is comprised of (1) end-user input data to ML models, (2) model features, (3) model outputs, and (4) the designers' derived data from model outputs. Further, this space can be projected (faceted) into interface data contexts and can include individual input data points, all data for a given end-user (persona), data-contexts that indicate temporality etc. Third, a data context can be bound to an interface state. Designers can evaluate the design for diverse users and contexts by generating interface previews for different data contexts. When prototyping for AI features, tools should allow designers to navigate across this design space flexibly. Designers should be able to switch between model simulation, design, test, analysis, and revision and repair.

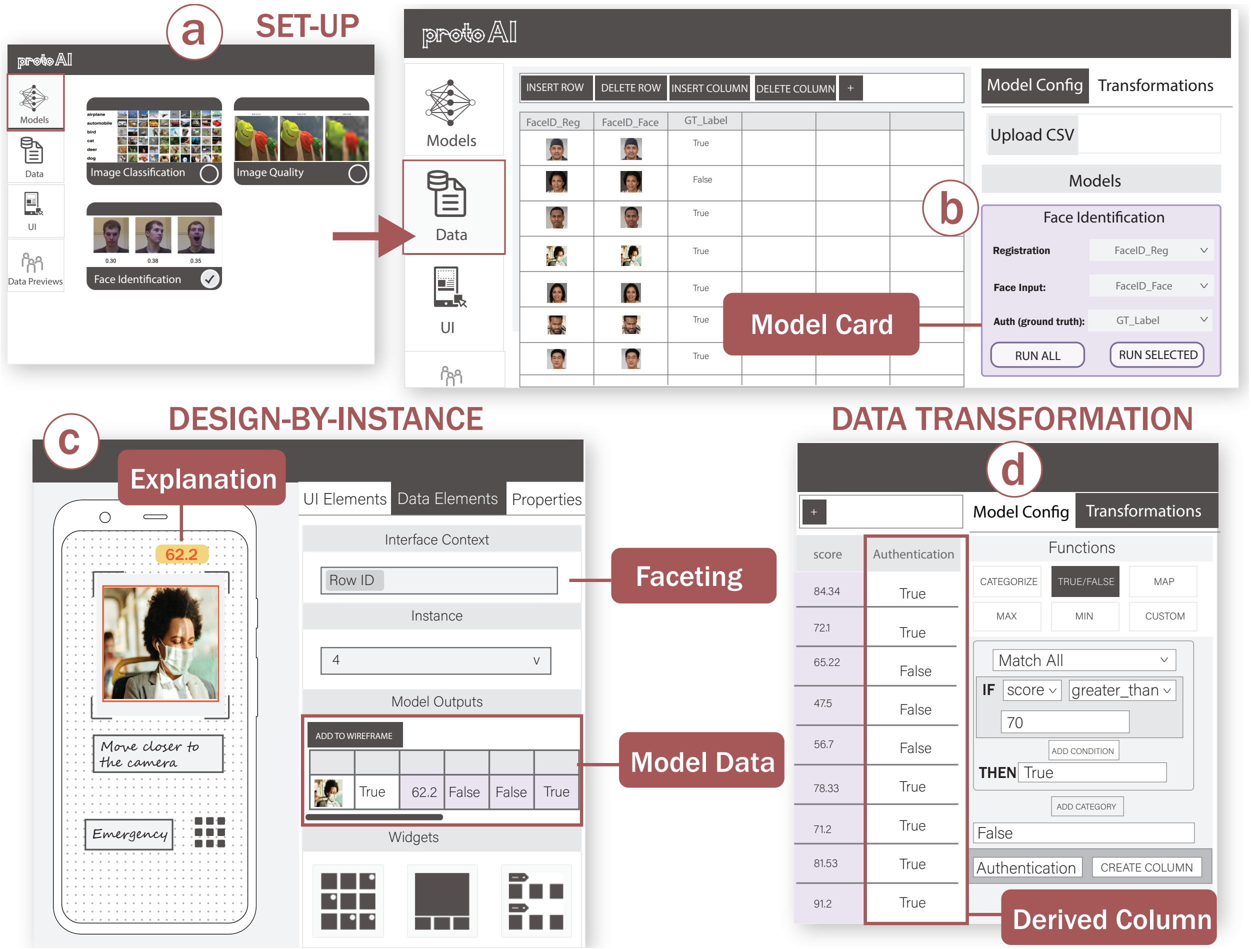


Figure 6.3: ProtoAI's user interface and features for MIP. To set-up, the designer (a) selects the Face-ID model, and (b) configures it using the model card. In the User Interface tab, the designer (c) incorporates model inputs and outputs in the wireframe; and (d) transforms Face match score into Boolean column in the Data tab.

6.3 Model-Informed Prototyping

Based on design considerations, we implemented ProtoAI to prototype AI-powered interfaces for AIX design. ProtoAI consists of four main views: (1) an AI models and services view (these can be implemented AI services or models, or Wizard of Oz 'stubs'), (2) a data view to import diverse input data for model simulation, (3) a UI 'designer' view to visually construct the interface prototype, and (4) a data previews view to simulate the interface design across different input data contexts. To better understand how a designer might use ProtoAI to engage in MIP, let us follow Divya, an AIX designer who is prototyping a Face-ID-based phone unlock experience.

6.3.1 Set-Up

Divya opens the ProtoAI application in the web browser. The Models tab is open by default and shows all of the AI services and models that are available in the system (Figure 6.3a). Divya's company has already assigned an engineering team to the project, and they have been working on an initial version of the Face-ID model. Divya selects the company's Face-ID model and navigates to the Data tab. The Data tab will allow her to import input data for different personas and scenarios of use. As shown in Figure 6.3b, the Data tab consists of a main editable spreadsheet view and sidebar view for model configuration. The spreadsheet can consist of *input data columns*, *feature/parameter columns*, *AI output columns*, and *derived (calculated) columns*. Column types are made distinct through color-coding. The sidebar view shows a model card [175] for each model selected. From the Face-ID model card, Divya sees that the model requires images (both for training/registration) and optional ground truth labels.

Based on her user research, Divya has curated a set of personas and portrait photos for each persona taken across different usage context (e.g., low light condition, crowded subway, person with a beard, facial hair, different skin tones, etc.). Divya can manually input data into the spreadsheet or import it from external sources (e.g., a CSV file). To simulate the model with this data, Divya maps the column headers on the spreadsheet with the model card inputs by selecting from a dropdown list of all columns. These images become the input data columns. Once configured, Divya runs the Face-ID model for the imported data (aligned with design consideration D1). ProtoAI extends the spreadsheet and appends additional columns with model outputs. The model output columns are color-coded to match the model configuration card. In an alternate scenario, in the absence of a pre-existing model, Divya can use the spreadsheet view to "draft" desired model behavior and outputs for different input data and share those specifications with her engineering team. The Face-ID model that her engineers have created return additional details: a percentage match score (calculated based on face distances in the face embedding space), an explainable heatmap rendering of the input image [210], and a set of Boolean flags for model features (e.g., whether a face was detected, eyes were closed, etc.). Using this simulated output, Divya can proceed to design the user interface for Face-ID based unlocking.

6.3.2 User Interface Design

Divya selects the User Interface tab, which consists of a design canvas and a sidebar for interface elements. The design canvas starts with a default phone template, but Divya can select others if needed (e.g., desktop or tablet). The sidebar consists of three panels, including the *UI Elements* panel which had a set of standard interface elements, the *Data Elements* panel which hosts input and model output data and a collection of widgets for MIP, and a *Properties* panel to set element-specific properties. To design the phone unlock experience, Divya wants to show the camera view in full screen, along with a button at the bottom for emergency calls and an icon on top to indicate the phone is scanning for a face. Divya first adds the emergency button by selecting the button element from the UI elements tab. She also adds a placeholder image on top of the screen to represent scanning status.

Next, to engage in design-by-instance prototyping, Divya opens the data elements panel (Figure 6.3c). This panel consists of a faceting control to set the wireframe's *data context* and a table showing the faceted data itself (a subset of the main spreadsheet view). The data context is the scope of end-user data that will be bound to the interface at runtime. The faceting feature is flexible and can set the data context to a single row or a set of rows nested and grouped by column names. For example, in a different scenario, Divya can set the data context to all images a persona has taken (e.g., for a photo album interface). Because the Face-ID UX shows the camera feed from the front-facing camera (i.e., a person's face), Divya sets the data context to a single row.

From the faceted table, Divya selects the cell value with the persona's face image and clicks on the 'Add to Wireframe' button. ProtoAI adds the image of the person's face onto the template, and Divya can adjust it to fit her design. Divya also adds the percentage match score value from the Face-ID model's output to the interface (from design consideration D2). While not intended for the final deliverable, Divya can use it to test and debug the interface design. To indicate this to ProtoAI, Divya toggles the 'set as explanation' flag in the properties tab for the score element. This will allow her to selectively show the explanation overviews later in the previews tab. For other complex layout needs, Divya can select entire columns, or brush select the desired data from the data table and add them to the interface as a widget. Based on AIX interface design patterns, ProtoAI implements an initial set of widgets for binding Boolean values to images, categorizing items by tags, and showing ranked order of items. Each widget has a predefined layout and can be bound to selected data along with explanation overlays for designers. The widget library can be extended in the future to support additional layout design needs.

6.3.3 Design Evaluation

At this point, Divya has an initial wireframe of the phone unlock interface designed using the portrait image from a single persona. She selects the Data Previews tab to evaluate her current design against different personas and their photos. ProtoAI automatically instantiates the screen interface based on the data context and using all data imported in the data tab (D4). As shown in Figure 6.1, the Data Previews tab consists of a scrolling grid view of the UI rendered for different users and their portrait photo variations. The preview view allows Divya to rapidly evaluate her design as it is being created and conduct design checks. In a different scenario, to evaluate model functionality

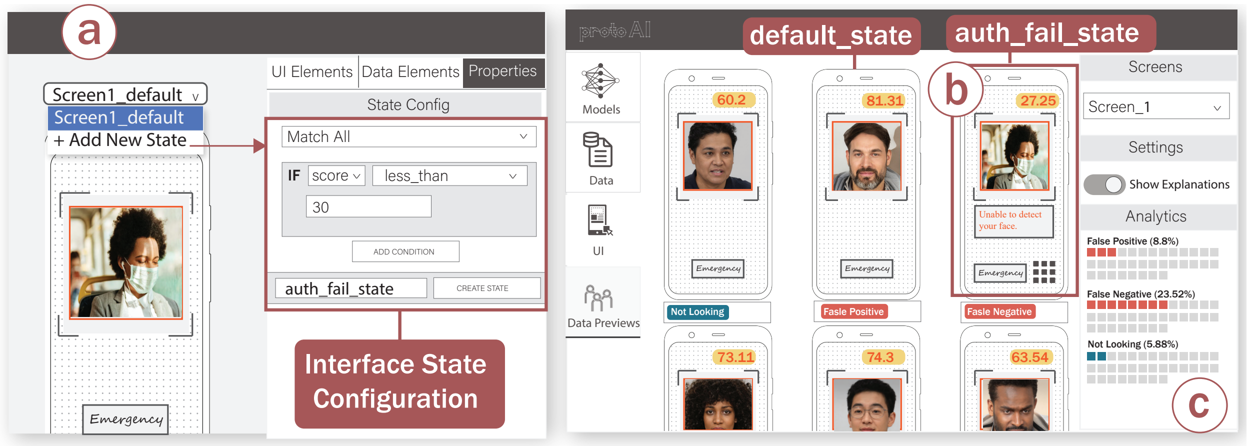


Figure 6.4: To design scenarios with Face-ID authentication failure, the designer (a) creates a new interface state conditioned on the match score below 30. In the Data Previews tab, ProtoAI generates appropriate interface states based on scores, and (c) provides analytics for number of instances with errors.

over time with model learning, Divya can configure data for different personas by providing different amounts of input data. This will allow Divya to visually see how the AI-powered interface responds after differing degrees of use. Divya can also check her design for different data sizes from model output (e.g., recommendations), ranging from no recommendations, a few recommendations, to tens of recommendations. Third, Divya can also evaluate the design for localization by providing inputs in different languages. Fourth, suppose the model's parameters require tweaking (such as the number of clusters). In that case, Divya can configure the data with different cluster sizes and compare the results in the data previews view.

6.3.4 Analysis, Revision, and Repair

ProtoAI's 'evaluation through previews' is intended to support the designer in analyzing design breakdowns in differing real-world contexts. Using the Data Previews view, Divya can iteratively revise the design to make it robust for a wider variety of users and contexts. ProtoAI offers a number of analysis features to support this iterative MIP workflow. Because Divya specified ground truth data for each of the photos, ProtoAI automatically compares the ground truth (Face-ID match) with model predictions and tags instances of false positives and false negatives. In the sidebar, ProtoAI provides a summary of each tag indicating the number of instances with that tag. Divya can see that in 16% of data, the model predicted an identity mismatch when the image was, in fact, the persona (i.e., false negatives). By checking the 'show explanations' flag in the sidebar, Divya can see the match score element she added in the UI tab. Similarly, Divya can also overlay other model factors and outcome values such as identified facial features or saliency maps to help her understand what the model computed from the image. Divya can also add her own tags to indicate domain-specific types of breakdowns or repairs. In this example, Divya sees that the model fails when the person is farther away or when their eyes are closed.

To address this issue, Divya switches back to the Data tab and creates a new calculated Boolean column that is set to 'true' if the face is not detected or when eyes are closed (D3). The Data tab allows for several different types of data *transformations*, including the categorization of numerical values (e.g., high, medium, and low), mapping transformations of model-assigned labels and values to end-user-friendly labels, calculating the minimum and maximum values, and custom formula functions for excel-like computation by specifying column headers and cells values to include in the computation (see Figure 6.3d). Through these transformations, Divya can design the model's API. Put another way, she controls the format in which the model output is presented in the user interface.

After creating the Boolean column, Divya returns to the UI tab to address the false-negative instances (D5). In ProtoAI, each screen can be assigned different screen states dependent on model behavior and values. Divya adds a new interface *state* to the unlock screen conditioned on the Boolean column value, which she configures using the properties panel (Figure 6.4a). In this state, Divya adds a message at the top of the screen prompting the user to move closer to the screen. In addition, Divya adds a numeric password option to address the remaining failure cases. When she returns to the previews view, Divya sees that instances of false negatives have the prompt message she just created. For interfaces with multiple screen states and screen-to-screen flow, ProtoAI offers a summary view showing a navigation diagram indicating how each end-user (based on their data) would engage in the AI-powered task flow. This allows Divya to see the probabilistic nature of the AI's task-flows to ensure that all users meet the desired goals. In this manner, Divya can design the interface using direct outputs from the ML model, evaluate her design against different AI and real-world constraints, and iteratively revise the design and repair the API to prototype the AI user experience (D5).

6.3.5 Implementation

We implemented ProtoAI as a web-based application using a client-server architecture. The server was written in Python and hosts different ML models. We use the metadata format in RunwayML [176] to specify the inputs and outputs to the model. Through this metadata, we generate the client-side model cards. This allows a number of already available models to be used in ProtoAI. The client is implemented using HTML and JavaScript. We make use of third party libraries for the UI design canvas [77], the spreadsheet interface [116], and formula parsing [38].

6.4 Design Scenarios

To demonstrate the utility of ProtoAI in operationalizing HAI design guidelines, we offer three usage scenarios based on real-world examples.

6.4.1 Social Media Feed–Automated Image Cropping

ProtoAI supports testing for, detecting, and fixing context breakdowns during design. A recent example of this need is the image auto-cropping feature offered on social media feeds (e.g., Twitter's problem [113] and their response [159]). We imagine a process by which ProtoAI can be used to fix the bias in cropping. The designer begins by collecting various images with different sizes, aspect ratios, salient points, and content semantics. They curate this data based on user research on photos uploaded to social media. The designer can then run the Auto-Cropping model against those images to view the cropped image within the user interface design. In the Data Previews tab, the designer sees that some images are cropped appropriately, but others leave out foreground objects or show only background. To investigate this issue, the designer can overlay the image saliency map (class activation mapping) returned by the model. Back in the preview mode, the designer sees that cropping fails when there are multiple salient points and no salient points. By tagging the images with appropriate labels, the designer sees that around 30% of images in the dataset have multiple salient points, and 5% have no salient points. To suggest a fix, the designer proposes an image widget for users that pans between different salient points in a loop. To resolve images with no salient points, the designer adds interface functionality for *manual* cropping using the AI-generated crop region as a suggestion.

6.4.2 Movie Recommendations–Changes with Use

Another challenge for designers is knowing how the interface and user experience may change over time as the AI learns from end-user data and feedback. ProtoAI can help simulate design previews over time and use. For example, we imagine a designer using ProtoAI to design a movie recommendation page. The designer can set up the data for different personas (either real-world preference or simulated)[1]. Based on input data, the model returns a set of movie recommendations for each persona and factors explaining why the movie was recommended. The designer then wireframes an initial interface listing all of the movies recommended by the AI. The previews tab shows that for personas with few or no input data about movies already watched, the recommendations do not align with the fictional persona's preferences. By looking at the confidence

[1] An alternative strategy for the designer would be to duplicate the persona rows and remove data. This simulates earlier versions of the persona with less training data

score for recommendations, the designer creates a new screen state for low confidence recommendations: instead of showing the movies, it asks end-users to select movie genres of interest. For other personas with sufficient input data and suitable recommendations, the designer chooses to present a categorized output by transforming confidence scores into confidence categories. Further by looking at the explanation factors, the designer can incorporate model explanations in text form: "Because you watched [explanation value], we think you might like:" Through data personas with differing inputs, ProtoAI allows designers to simulate model behavior over time of use, and design appropriate interface experiences.

6.4.3 Chat Assistant–Mixed Initiative Design

A guiding principle for integrating AI capabilities into task workflows is to determine the utility of the AI for end users [120]. If the AI is confident about the end user's goals, it can tend towards automation. If the goal can be resolved with minimal support from users, the AI can engage in a mixed-initiative dialog with end-users. In all other cases, AI should not automate the task. This design profile for HAI applies to a variety of AI-powered application designs; yet for designers, understanding the utility function is challenging. The design-by-instance, and Data Preview features in ProtoAI can help designers achieve the mixed-initiative design. For instance, consider a chat assistant's design that prompts end-users with task actions based on chat messages. When the user comments, "Let's meet at Bob's Burger place," the AI can pull up directions to the location if confidence is high. If there are multiple outlets, the AI can present a list sorted by proximity and ask the end-user to pick a specific location. In other cases, the AI should ask the end-user to manually input the address. In ProtoAI, the designer can curate such chat messages for the ML model. In the Preview tab, they can tag instances with incorrect recommendations and overlay each recommendation's confidence score. Later, they can create different screen-states for mixed-initiative and manual inputs using confidence score thresholds. The preview section allows the designer to experience first-hand the subjective utility for end-users and then offer necessary adaptations to their interface design.

6.5 Preliminary Evaluation

To gather feedback on ProtoAI's implementation of MIP, we conducted a preliminary online user study. We aimed to (1) assess whether designers can successfully leverage ProtoAI's features for prototyping AI-powered interfaces, and (2) collect feedback on the model-informed prototyping workflow. We recruited 10 UX designers for the study with expertise in prototyping user interfaces using off-the-shelf tools such as Figma, Sketch, Axure, and Adobe XD. Six participants had prior

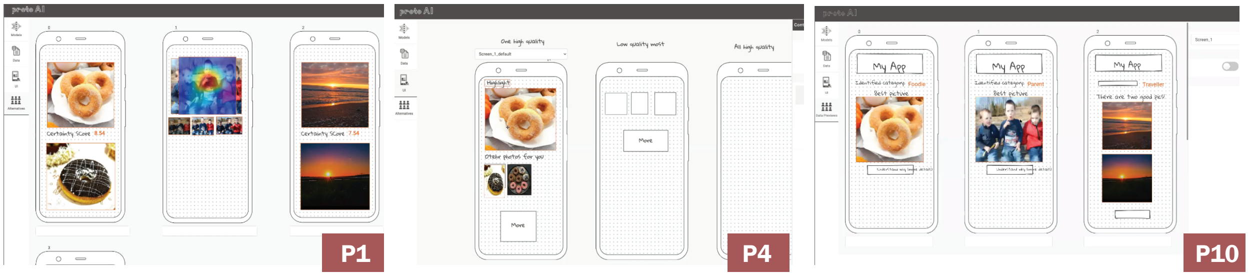


Figure 6.5: Example designs generated by our study participants for AI based photo recommendation for Instagram.

experience designing AI-powered applications. Each session lasted 75 minutes, and participants were compensated with $20 for their time. At the start of each session, we provided an overview of MIP. We gave an in-depth walkthrough of ProtoAI features using image classification as an example.

Following the walkthrough, participants engaged in a design activity using ProtoAI. We asked participants to design an AI-powered interface to recommend images (from a set) to upload to Instagram. In the interest of session time, we provided data consisting of four personas with five images for each. Each image included a quality score ranging from 0-10 (higher scores indicating better quality) based on a Neural Image Assessment model [230]. We selected images such that different personas had different score ranges and variations in the differences between scores. For each image, we also generated a class activation heatmap [210] to probe participants on the explainability features of ProtoAI. Participants launched ProtoAI on their web-browser and shared their screen during the session. We asked participants to think-aloud during this phase of the session and recorded them. Once participants completed the task, we conducted a freeform interview to understand how they would use ProtoAI for AIX problems they had worked on in the past, provided feedback on ProtoAI's features and interface, and commented on MIP workflow. At the end of the study, participants filled out a usability questionnaire [160] and the NASA-Task Load Index questionnaire [103].

6.5.1 Findings

6.5.1.1 Model-Informed Prototyping with ProtoAI

Across all sessions, designers created an image recommendation UX with one or more screens (Figure 6.5). They used data previews as they worked and created new screen states based on the generated previews. In five of the sessions, designers directly started the activity using data elements, including persona images and quality scores. For instance, in session 10, the designer crafted an initial layout showing only the image with the highest score. Then, by looking at the data tab, they realized there were some small differences between images with the second and third highest scores. They then created two new derived columns to compute the differences in scores and created new screen states for the best two and three images. Designers also used the 'categorize transformation' function to bucket image scores into high, medium, and low categories. In the remaining five sessions, designers first created placeholder layouts and then imported data from the data elements tab. In session 4, the designer created a set of placeholder screens for different data conditions, including one high-quality image, all low-quality images, and all high-quality images. They then replaced the placeholder with real data and model outputs. All designers made use of the explanation overlays. By looking at the CAM heat maps, they revised their designs in ways we did not anticipate. For instance, in session 1, the designer created a new task-flow for end-users to crop the image based on salient regions indicated by the heatmap and re-compute the quality score. In session 5, the designer suggested addressing tasks with no high-quality images by showing the CAM view to end-users and allowing them to retake the photo. Based on the NASA-TLX questionnaire, participants rated the overall task workload of 50.3 ($SD = 12.49$). A breakdown of individual components show that participants ratings were: *mental demand*: 59 ($SD = 18.07$); physical demand: 24 ($SD = 21.53$); temporal demand:38 ($SD = 20.70$); performance:38.5 ($SD = 18.86$); effort:49 ($SD = 12.2$); and frustration:37 ($SD = 16.36$).

6.5.1.2 Utility of MIP

Designers with prior experience designing AI-powered applications and knowledge of HAI guidelines (n=6) saw value in MIP (and ProtoAI). They all mentioned their current data-driven design workflow either by writing code or analyzing data using spreadsheets. In particular, designers appreciated the data-to-interface pipeline through model simulation, auto-generated data previews, and carrying out data transformations during the design process. In providing feedback about the overall workflow, P4 commented: *"Right now I will have my hypothesis about the data and go back to the engineer and ask them to give me the output, but they say that those data instances will not occur, there is a lack of transparency, and there are layers of gates I need to get through before I can do the next step. This tool makes it easy for me to carry out the entire flow on my own."* When commenting about prototyping for data instances, P5 commented: *"The hardest thing about designing for AI is getting the right data. You can make something look good with fake labels and 'ipsum-lorem,' but using real data to mock things up helps you see where things are broken. I think automatically generating the alternatives using the data is very powerful."* For participants new to AIX design, they compared MIP to their current workflows. They commented they needed scaffolding to understand the AI model and outputs and incorporate data elements in their design.

6.5.1.3 User Experience

Overall, participants found ProtoAI's interface intuitive and easy to use. They appreciated the flexibility and connectedness of end-user data across different tabs (Data, UI, and Previews). P1 commented that ProtoAI is beneficial at the brainstorming stage, where instead of wireframing on the whiteboard, they can quickly input data and desired model output and test out interface alternatives. P8 commented on the explanation overlays, stating they can add model outputs on the interface and flexibly include it in the final design or flag it as "explanation for the designer." Participants made suggestions for section and tab labels, which we incorporated into the final design (e.g., in the prototype used in the study, 'data previews' tab was labeled 'alternatives'). They also recommended having pop-out windows for the data elements tab to avoid scrolling across each row. Based on the usability questionnaire, on a seven-point scale, participants rated ProtoAI's to be easy to use ($mean = 5.88$,$SD = 0.9$), and flexible ($mean = 5.63$,$SD = 0.72$). Participants rated their learnability (i.e., can learn it quickly) a mean score of 5.33 ($SD = 1.65$), and learning without written instructions as 3.22 ($SD = 1.39$). In future iterations, we can support on-boarding through guided walkthroughs and use cases of design guidelines. Encouraged by the overall feedback, we plan to conduct a comparative evaluation of ProtoAI against commercial prototyping tools and assess the quality of design output using ProtoAI.

6.6 Discussion and Future Work

To design user interfaces for AI-powered applications, designers need access to the underlying AI. Therefore, digital prototyping tools should escape the 'black-box' view of AI by incorporating the AI model's characteristics into the UI prototyping process. In this work, we define a new paradigm for UX design for AI-powered applications, which we call AIX. To accomplish AIX design, we have demonstrated how ProtoAI's implementation of *Model-Informed Prototyping* allows designers to (1) directly incorporate an AI's output into their design, (2) test their design across different input data contexts, and (3) iteratively assess and adapt their interfaces for explainability, failure, and model feedback. Based on our evaluation and participants' feedback, ProtoAI allows designers to prototype AI-powered UI, provide just-in-time model simulation and outputs without AI model engineering, and transform model outputs to meet interface presentation needs. In addition, the data-level representations in ProtoAI correspond to engineering representations of the AI service's API. This affords opportunities for communication, negotiation, and co-design between designers and engineers. Specifically, future work can investigate how AIX designers can drive AI model parameters based on interface features, negotiate model features and outputs necessary for explainability, and communicate discovered failure instances with engineers for model improvement.

End-user data is a critical aspect of MIP. In this regard, ProtoAI offers flexibility for designers to manually input data from user research and simulated data to explore design their hypotheses about AI behavior. Besides, they can directly import data from other human-centered design processes (e.g., Data-Assisted Affinity Diagramming [222]). However, we do not investigate specific data generation needs during the prototyping process in our current work. When prototyping, designers may need access to diverse data to consider both success and failure cases at the AI and UI levels. We are currently exploring ways to support synthetic data generation needs through expressive queries. For instance, visualization design tools allow designers to generate data with specific statistical and visual properties [168, 94]. AIX designers may also need to work with sensor data or implicit feedback collected by system logs. Future work should look at ways to support these specific data and analysis needs and advanced user-modeling for MIP. Third, ProtoAI has the potential to support Responsible AI needs such as fairness, accessibility, and transparency. AI engineers are asked to evaluate their data and ML models for responsible AI criteria (e.g., AI Fairness 360 [25]), and AIX designers can use tools like ProtoAI's data previews to detect interface failures in responsible AI design.

In ProtoAI, we assume that MIP is useful during early-stage prototyping (i.e., generative wireframing). This allows us to trade-off design complexity for detailed data. Further, while ProtoAI supports evaluation by designers through data previews, certain types of experiential design failures may not be apparent to designers. Future research should look at how MIP can be integrated into later stages of AIX prototyping and usability testing workflows. This includes supporting interactive and click-through prototypes, sharing prototypes with end-users, and logging capabilities. Finally, as pedagogy and practice of AI application design continues to evolve, we envision AIX tools like ProtoAI will enable students and novice designers to develop necessary skills for AIX prototyping. We imagine a library of widgets implementing AIX design patterns and explainable overlays to scaffold designers' learning process.

6.7 Summary

While AI capabilities are prevalent in everyday and high-stakes software applications, end-users frequently encounter unpleasant AI experiences. A challenge for designers is that their current design tools mainly assume a 'black-boxed' view of AI. This restricts the designer's ability to anticipate and address breakdowns in AIX. To maximize end-user success with AIX, designers should directly work with underlying AI features during the design process. In this work, we present *Model-Informed Prototyping*, a workflow that interleaves AI exploration and UI prototyping tasks. Our implementation of MIP, ProtoAI, allows designers to directly invoke AI models and services, incorporate model outputs into interface design, and iteratively and rapidly evaluate their design choices across diverse end-users and their data context. We demonstrate how ProtoAI can support designers in operationalizing best practice HAI guidelines. Preliminary feedback from designers highlights ProtoAI's potential to empower designers by providing them just-in-time access to AI features.

CHAPTER 7

Conclusion and Future Work

In this dissertation, I argue that both AI-first and UX-first approaches to HAI design are problematic. In both workflows, designers and engineers experience knowledge blindness about end-users and AI's capabilities and limitations. Disregarding knowledge gaps with the assumption that the other expert will 'adapt' leads to premature design specifications, frustration for software teams, and AI-UX mismatches creating gulfs in human experiences with AI. As an alternative, I propose a collaborative approach in which AI and UX design are treated as one, i.e., AI experience design. To operationalize this view, I investigate current collaborative practices, identify means to bridge knowledge barriers, propose a co-design process model, and develop data-driven tools for creating AIX.

7.1 Summary of Contributions

Based on the qualitative studies and the support tools I have built, I make the following contributions towards designing AI experiences:

- In Chapter 2, I synthesize work across HCI, AI, Software Development, and Sociology to motivate my study of collaborative design of HAI. I identify challenges in human-centered AI arising from conventional software practices, including modular design, abstractions, and separation of concerns. Building on this understanding, I summarize current AI-UX workflows, characterize expertise and design challenges, and review the roles of boundary objects and data in collaborative design.
- To examine the gaps in AI software workflows and current practices in the industry, in Chapter 3, I conduct interviews with practitioners across HCI and AI roles. I contribute a component model representation of AIX derived from my analysis of design guidelines. I report on friction and challenges at the AI-UX boundaries and identify the critical role of

"leaky" abstractions and boundary representations in bridging knowledge boundaries. Further, I recommend deferred specification as means for AIX design through iterative vertical prototyping and constant evaluation.

- Based on insights about boundary artifacts, in Chapter 4, I characterize the interactions between AI and HCI practitioners when co-creating AIX through an in-lab design study. From observing collaboration strategies, I identify the crucial role of end-user data as the "lingua franca" (i.e., content common ground) between designers and engineers. By using data probes, designers construct designerly proxies to specify AI needs. Further, data probes facilitate divergent thinking, design convergence, and validation. Based on these findings, I propose a process model for collaborative AIX design and offer considerations for incorporating data probes in AIX design tools.
- In Chapter 5, I propose Data-Assisted Affinity Diagramming (DAAD) for combined analysis of qualitative and quantitative end-user data to generate nuanced personas for defining AI's training data needs. By developing an augmented-reality-based prototypical tool, Affinity Lens, I evaluate DAAD through multiple lab studies with datasets on eating habits of undergraduate students and self-reported privacy concerns on social media sites. Through Affinity Lens, I contribute an approach for creating data-driven personas for AIX design.
- Finally in Chapter 6, I propose an interface prototyping workflow for AIX called Model Informed Prototyping (MIP). MIP interleaves AI exploration and UI prototyping tasks to support designers in operationalizing best practice HAI guidelines. ProtoAI, a prototypical tool for MIP, allows designers to directly invoke AI models and services, incorporate model outputs into interface design, and iteratively and rapidly evaluate their design choices across diverse end-users and their data context. Preliminary feedback from designers highlights ProtoAI's potential to empower designers by providing them just-in-time access to AI features.

Collectively, this work provides solutions to identified problems in combining expertise from AI and HCI to operationalize the goals for HAI.

7.2 Designing AI Experiences - An Integrated Walkthrough

Based on the findings in Chapters 2- 6, here I offer a walkthrough of co-designing AI experiences with designers and engineers. I intend to underscore and summarize my thesis that, unlike conventional software practices that favor clear separation of concerns, creating AI experiences with designers and engineering practitioners benefit from (1) "leaky" abstractions to share information across different layers of the application, and (2) delayed specifications through vertical prototyping, and (3) constant evaluation using data tools.

Let us imagine designing an AI-powered experience to support recording bird-sightings (e.g., Merlin Bird ID [53]). At the start of the project, by taking a human-centered approach, the designer will conduct user research with bird-watchers (including expert ornithologists and novice hobbyists) to understand current practices for logging bird sightings. In addition to interviews with bird-watchers, the designer collects past log data of bird-sightings and example photos that participants are able to share. They may also gather survey responses from participants about their experience, expertise, needs, and challenges in logging bird-sightings. Using this mixed data from interview notes, survey responses, and log data, the designer conducts DAAD using Affinity Lens. The outcome is a set of *data personas* including that of a student interested in Avian Science, a seasoned hiker, an official at a natural reserve, an expert ornithologist, and an urban bird watcher.

Using the data personas as a starting point, the designers and engineers engage in a preliminary co-design session as described in Chapter 4. By brainstorming about potential AI-infused scenarios for each persona, the designers and engineers identify potential AI behavior such as recognizing the sighted bird species, recalling past sightings of the same bird, and highlighting distinguishing characteristics of individual bird species. One scenario might include an official at a nature reserve tracking recovery of an injured bird or the nesting behavior of a pair of birds. Next, the designers and engineers conduct cognitive walkthroughs using the scenarios to determine specific implementation logic, necessary features for training data, and assumptions and constraints for AI design. For instance, based on inputs from expert ornithologists, teams may determine that in addition to visual characteristics, experts also use bird sounds (such as a bird chirping) and movement information to identify the bird. They consider these additional attributes when designing the bird identification AI. Using the behavior and implementation details, the AI engineer will create model cards documenting API details, including inputs and outputs to the AI. With this information, the designer prototypes lo-fidelity user interfaces by incorporating data probes from individual personas. At the end of this generative process, designers and engineers agree on a high-level solution and approach for the AI-powered bird logging experience through negotiation and validation.

In subsequent iterations, as described in Chapter 3, the designers and engineers continue to envision and evolve their design specifications to reach a consensus that aligns with the needs of their target personas. By embracing abstraction leaks, the engineers construct visual dashboards, computational notebooks with data queries, and spreadsheets to share details about AI's training data characteristics. This includes data about species distribution, resolution of images and noise, image composition including birds at flight, a flock of birds, and close-up shots. Using this information, the designer evaluates the data for representativeness. Further, the designer and engineers may work with expert ornithologists to annotate the training data for species ID, distinguishing visual features, and other details to support explainability. To facilitate the annotation process, the designer may develop code-books with annotation guidelines and examples and create usable data annotations tools to capture the required attributes for AI's training accurately and in a user-friendly manner.

During this iterative process, the designer also engages in model-informed prototyping using ProtoAI. By integrating with the ML models under development and data from individual personas, designers prototype and assess their interface designs for AI's uncertainties. Through this prototyping process, designers may determine the right balance between automation and augmentation, create mixed-initiative widgets for end-user feedback, and design explainability features to teach novice bird watchers about different bird species. For example, using data from model outputs, designers may prototype explainability alternatives such as categorized confidence scores, heat maps showing parts of the image that correspond to established features for the species, and textual descriptions and annotations about the species. Engineers also revise their model and API design to support the correct input and output formats for presentation and negotiate specific feedback needs for AI's learnability. The result is an AI-powered experience for logging bird sightings that meets end-user needs and their expectations from AI.

7.3 Limitations

In this body of work, I employ interviews, in-lab design studies, and system design and implementation as the primary methods of inquiry into collaborative HAI practices. While the qualitative studies with practitioners provided insightful findings, the specifics of low-level details and *in-situ* interactions are difficult to glean without direct observation of work in practice. For instance, due to non-disclosure agreements, interview participants could not directly share product artifacts they mentioned during the interviews. Consequently, my findings may not capture domain- and data-related nuances in the conceptualization of boundary representations. Second, the evaluation of the process model and data tools are primarily conducted in controlled settings such as in-lab and in the classroom. Evaluation in the wild may offer a comprehensive understanding of its utility for practitioners. Third, much of today's AI is implemented using labeled datasets and supervised learning. Therefore, in this dissertation, I primarily focus on data-driven methods for AIX design. However, I do not consider other rule-based and knowledge-based AI techniques which may produce differing findings. Finally, implementing my conclusions about collaboration practices from this dissertation in practice settings would require changes to work team structures and incentives within organizations producing HAI. However, my work does not study the effort, cost, and bottlenecks involving in making the recommended changes towards co-designing AIX.

7.4 Future Work

The space of human-centered AI is vast. The processes and tools presented in this dissertation offer valuable directions for future work in collaborative tools, responsible AI design, and AIX pedagogy.

7.4.1 Comprehensible Representations and Leaky Abstractions

A fundamental affordance of leaky abstractions is allowing individuals with differing expertise to *co-construct* designs. Through information sharing, design manipulation, re-representation, gap-filling suggestions, and feedback, diverse stakeholders can collaboratively produce design solutions. Future research should investigate distinct characteristics and needs for leaky artifacts across different domains and AI techniques, and then develop tools to support them. For instance, in the process-model study, engineers annotated visualizations on top of interface prototypes to convey the design space for explainability. While useful, designers lacked the means to comprehend, reciprocate, and design with model explainability in complex cases. We need to invent new artifacts, visual representations, and tools to effectively support creating and sharing information leaks through such practices. As with cognitive dimensions [49] to evaluate API effectiveness, we also need new guidelines and attributes to define effective leaky abstractions.

7.4.2 Responsible AI Design

An essential aspect of AIX is to ensure the responsible design of AI-powered applications. During co-design, designers and AI engineers should carefully consider various responsible AI criteria, including transparency, accountability, accessibility, fairness, privacy, and security. Moreover, differing recommendations from these criteria require prioritization and trade-off assessment when making design choices. However, existing design tools and processes lack support for critical design thinking about responsible AI criteria bridging the boundaries between design, engineering, and deployment contexts. My previous work on ProtoAI offers initial insights about how designers can assess AIX alternatives across diverse users and their contexts of use. As new guidelines and recommendations regarding responsible practices emerge, future research should re-examine socio-technical practices and develop end-to-end strategies for fulfilling the needs of responsible AI. One specific direction pertains to AIX evaluation for Fairness criteria. This demands defining and aligning performance metrics across AI and usability through co-design. Co-design will require innovative boundary artifacts and 'leaks' to bridge metrics and techniques across AI and UX.

7.4.3 AIX Pedagogy

The insights from this dissertation have direct applicability for design and AI engineering pedagogy. Ideally, to reduce the knowledge blindness identified in this dissertation, AIX practitioners benefit from π shaped expertise across HCI and AI (i.e., in-depth understanding of HCI *and* AI) [23]. But acquiring such understanding is impractical given the rapidly advancing state-of-the-art in both AI and design. Instead, HCI pedagogy should equip future practitioners with data-driven design tools and methods to facilitate co-design. For instance, designers should receive training in constructing data probes for design, model informed prototyping, and understanding visual representations (e.g., interpretable ML) that occur at the boundaries. To support pedagogy, we need new toolkits and instructions to make AIX accessible to students from differing backgrounds. Similarly, AI engineers should receive training in the parallel processes between AI and UX design (i.e., the process model). They should be trained to understand the role of UX in AIX design and to work with designerly proxies to deliver boundary representations for collaboration. Finally, AIX curriculum should bring together students from varying backgrounds to engage in team co-learning. Multidisciplinary pedagogical initiatives are essential to shaping the future of AIX into practice.

7.5 Concluding Remarks

As a technical HCI scholar, I have created several AI systems across various domains, including human learning, sensemaking, and creativity. While off-the-shelf AI has offered a starting point to design (i.e., intelligence ”on-tap”), I have found that the design solution is less than optimal without ‘repairing’ the AI itself to fit design needs. The AIX design approach in this dissertation provides direction to create a more fitting AI from the ground up. The more concrete and precise the AI’s capability, and the more aligned with user needs, the better the AI can perform the task. Thus, in HAI co-design, uncertainties are minimized along with constraints and adaptations in user-facing components. My hope is that this dissertation will steer us away from the pursuit of general-purpose AI. Instead, my hope is to motivate AI and UX practitioners towards effective ways of creating human need-specific AIX solutions through collaboration and co-design.

BIBLIOGRAPHY

[1] Adobe. Adobe XD, 2020.

[2] Philip Agre and Philip E Agre. *Computation and human experience*. Cambridge University Press, 1997.

[3] Robert Akscyn, Elise Yoder, and Donald McCracken. The data model is the heart of interface design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 115–120, 1988.

[4] Christopher Alexander. *Notes on the Synthesis of Form*, volume 5. Harvard University Press, 1964.

[5] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. The state-of-the-art of set visualization. In *Computer Graphics Forum*, volume 35, pages 234–260. Wiley Online Library, 2016.

[6] Saleema Amershi. Guidelines for human-ai interaction design, 2019.

[7] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: a case study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, pages 291–300. IEEE Press, 2019.

[8] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120, 2014.

[9] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 337–346, 2015.

[10] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 3. ACM, 2019.

[11] Christopher Andrews, Alex Endert, Beth Yost, and Chris North. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization*, 10(4):341–355, 2011.

[12] Timo Arnall. Exploring'immaterials': Mediating design's invisible materials. *International Journal of Design*, 8(2), 2014.

[13] Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. Software engineering challenges of deep learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 50–59. IEEE, 2018.

[14] Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys (CSUR)*, 54(5):1–39, 2021.

[15] Jan Auernhammer. Human-centered ai: The role of human-centered design research in the development of ai. 2020.

[16] Julie Baca, Daniel Carruth, Elijah Davis, and Daniel Waddell. Merging the cultures of design and engineering: A case study. In *International Conference of Design, User Experience, and Usability*, pages 628–641. Springer, 2018.

[17] Balsamiq. balsamiq, 2021.

[18] Jack Bandy. Problematic machine behavior: A systematic literature review of algorithm audits. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–34, 2021.

[19] Thierry Bardini. *Bootstrapping: Douglas Engelbart, coevolution, and the origins of personal computing*. Stanford University Press, 2000.

[20] Solon Barocas, Anhong Guo, Ece Kamar, Jacquelyn Krones, Meredith Ringel Morris, Jennifer Wortman Vaughan, Duncan Wadsworth, and Hanna Wallach. Designing disaggregated evaluations of ai systems: Choices, considerations, and tradeoffs. *arXiv preprint arXiv:2103.06076*, 2021.

[21] Sumit Basu, Danyel Fisher, Steven M Drucker, and Hao Lu. Assisting users with clustering tasks by combining metric learning and classification. In *AAAI*, 2010.

[22] Patrick Baudisch and Ruth Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 481–488. ACM, 2003.

[23] Eric PS Baumer. Toward human-centered algorithm design. *Big Data &amp; Society*, 4(2):2053951717718854, 2017.

[24] Michel Beaudouin-Lafon and Wendy E Mackay. Prototyping tools and techniques. In *Human-Computer Interaction*, pages 137–160. CRC Press, 2009.

[25] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[26] Jesse Josua Benjamin, Arne Berger, Nick Merrill, and James Pierce. Machine learning uncertainty as a design material: A post-phenomenological inquiry. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021.

[27] Paul Beynon-Davies and Steve Holmes. Design breakdowns, scenarios and rapid application development. *Information and software technology*, 44(10):579–592, 2002.

[28] Eric A Bier, Maureen C Stone, Ken Pier, William Buxton, and Tony D DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM, 1993.

[29] Melanie Birks, Ysanne Chapman, and Karen Francis. Memoing in qualitative research: Probing data and processes. *Journal of research in nursing*, 13(1):68–75, 2008.

[30] Woodrow Wilson Bledsoe. The model method in facial recognition. *Panoramic Research Inc., Palo Alto, CA, Rep. PR1*, 15(47):2, 1966.

[31] Eli Blevis, Youn-kyung Lim, Erik Stolterman, et al. Regarding software as a material of design. In *Proceedings of Design Research Society International Conference*, pages 1–18, 2006.

[32] Barry Boehm. Requirements that handle ikiwisi, cots, and rapid change. *Computer*, 33(7):99–102, 2000.

[33] Erin Brady, Meredith Ringel Morris, Yu Zhong, Samuel White, and Jeffrey P Bigham. Visual challenges in the everyday lives of blind people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2117–2126. ACM, 2013.

[34] Jacob T Browne. Wizard of oz prototyping for machine learning experiences. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, page LBW2621. ACM, 2019.

[35] Louis L Bucciarelli. Between thought and object in engineering design. *Design studies*, 23(3):219–231, 2002.

[36] Louis L Bucciarelli and Louis L Bucciarelli. *Designing engineers*. MIT press, 1994.

[37] Marion Buchenau and Jane Fulton Suri. Experience prototyping. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 424–433, 2000.

[38] Krzysztof Budnik. Formula parser, 2020.

[39] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

[40] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[41] Carrie J Cai and Philip J Guo. 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*, volume 19, 2019.

[42] Paul R Carlile. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization science*, 13(4):442–455, 2002.

[43] Kim Carmona, Erin Finley, and Meng Li. The relationship between user experience and machine learning. *Available at SSRN 3173932*, 2018.

[44] John M Carroll, Robert L Mack, and Wendy A Kellogg. Interface metaphors and user interface design. In *Handbook of human-computer interaction*, pages 67–85. Elsevier, 1988.

[45] Marcelo Cataldo and James D Herbsleb. Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pages 65–72. ACM, 2010.

[46] M Ceconello, D Spallazzo, and M Sciannamè. Design and ai: prospects for dialogue. 2019.

[47] Senthil Chandrasegaran, Sriram Karthik Badam, Lorraine Kisselburgh, Karthik Ramani, and Niklas Elmqvist. Integrating visual analytics support for grounded theory practice in qualitative text analysis. In *Computer Graphics Forum*, volume 36, pages 201–212. Wiley Online Library, 2017.

[48] Herbert H Clark and Susan E Brennan. Grounding in communication. 1991.

[49] Steven Clarke. Describing and measuring api usability with the cognitive dimensions. In *Cognitive Dimensions of Notations 10th Anniversary Workshop*, page 131. Citeseer, 2005.

[50] Mangoslab Co. Nemonic mini printer, 2018.

[51] Melvin E Conway. How do committees invent. *Datamation*, 14(4):28–31, 1968.

[52] Eric Corbett, Nathaniel Saul, and Meg Pirrung. Interactive machine learning heuristics.

[53] Cornell. The cornell lab of ornithology, 2021.

[54] Intel Corporation. Open cv library, 2018.

[55] Design Council. The 'double diamond'design process model. *Design Council*, 2005.

[56] Henriette Cramer and Juho Kim. Confronting the tensions where ux meets ai. *interactions*, 26(6):69–71, 2019.

[57] Yanqing Cui, Jari Kangas, Jukka Holm, and Guido Grassel. Front-camera video recordings as emotion responses to mobile photos shared within close-knit groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 981–990. ACM, 2013.

[58] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *ACM SIGIR Forum*, volume 51, pages 148–159. ACM, 2017.

[59] Richard C Davis, T Scott Saponas, Michael Shilman, and James A Landay. Sketchwizard: Wizard of oz prototyping of pen-based user interfaces. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 119–128, 2007.

[60] Cleidson RB de Souza, David Redmiles, Li-Te Cheng, David Millen, and John Patterson. 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*, pages 63–71. ACM, 2004.

[61] David Dearman and Khai N Truong. Why users of yahoo!: answers do not answer questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 329–332. ACM, 2010.

[62] Dominik Dellermann, Adrian Calma, Nikolaus Lipusch, Thorsten Weber, Sascha Weigel, and Philipp Ebel. The future of human-ai collaboration: a taxonomy of design knowledge for hybrid intelligence systems. 2019.

[63] Norman K Denzin and Yvonna S Lincoln. *The Sage handbook of qualitative research*. sage, 2011.

[64] Marie Desjardins, James MacGlashan, and Julia Ferraioli. Interactive visual clustering. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 361–364. ACM, 2007.

[65] Kristin N Dew and Daniela K Rosner. Lessons from the woodshop: Cultivating design with living materials. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[66] Virginia Dignum. Responsible artificial intelligence: designing ai for human values. 2017.

[67] Dennis P Doordan. On materials. *Design Issues*, 19(4):3–8, 2003.

[68] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 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*, pages 278–288. ACM, 2017.

[69] Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing*, 4(4):18–26, 2005.

[70] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. *Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-efficacy*, pages 127–153. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[71] Steven M Drucker, Danyel Fisher, and Sumit Basu. Helping users sort faster with adaptive machine learning recommendations. In *IFIP Conference on Human-Computer Interaction*, pages 187–203. Springer, 2011.

[72] Susan Dumais, Edward Cutrell, Raman Sarin, and Eric Horvitz. Implicit queries (iq) for contextualized search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 594–594, New York, NY, USA, 2004. ACM.

[73] Maria R Ebling. Translucent cache management for mobile computing. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1998.

[74] Malin Eiband, Hanna Schneider, Mark Bilandzic, Julian Fazekas-Con, Mareike Haug, and Heinrich Hussmann. Bringing transparency design into practice. In *23rd international conference on intelligent user interfaces*, pages 211–223, 2018.

[75] Daniel Engelberg and Ahmed Seffah. A framework for rapid mid-fidelity prototyping of web sites. In *IFIP World Computer Congress, TC 13*, pages 203–215. Springer, 2002.

[76] Boris Ewenstein and Jennifer Whyte. Knowledge practices in design: the role of visual representations asepistemic objects'. *Organization studies*, 30(1):07–30, 2009.

[77] Fabric.js. Fabric.js html canvas library, 2020.

[78] Jennifer Fereday and Eimear Muir-Cochrane. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, 5(1):80–92, 2006.

[79] Ylva Fernaeus and Petra Sundström. The material move how materials matter in interaction design research. In *proceedings of the designing interactive systems conference*, pages 486–495, 2012.

[80] Figma. Figma, 2020.

[81] Gerhard Fischer. Symmetry of ignorance, social creativity, and meta-design. *Knowledge-Based Systems*, 13(7-8):527–537, 2000.

[82] David Flink. The wire: Your ai-powered 'to do' list, 2020.

[83] Johannes Fuchs, Roman Rädle, Dominik Sacha, Fabian Fischer, and Andreas Stoffel. Collaborative data analysis with smart tangible devices. In *IS&T/SPIE Electronic Imaging*, pages 90170C–90170C. International Society for Optics and Photonics, 2013.

[84] Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and Machines*, 30(3):411–437, 2020.

[85] Rolando Garcia, Vikram Sreekanti, Neeraja Yadwadkar, Daniel Crankshaw, Joseph E Gonzalez, and Joseph M Hellerstein. Context: The missing piece in the machine learning lifecycle. In *KDD CMI Workshop*, volume 114, 2018.

[86] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[87] Florian Geyer, Ulrike Pfeil, Jochen Budzinski, Anita Höchtl, and Harald Reiterer. Affinitytable-a hybrid surface for supporting affinity diagramming. In *IFIP Conference on Human-Computer Interaction*, pages 477–484. Springer, 2011.

[88] Elisa Giaccardi and Elvin Karana. Foundations of materials experience: An approach for hci. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2447–2456, 2015.

[89] Cristina B Gibson. From knowledge accumulation to accommodation: Cycles of collective cognition in work groups. *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior*, 22(2):121–134, 2001.

[90] Yolanda Gil, James Honaker, Shikhar Gupta, Yibo Ma, Vito D'Orazio, Daniel Garijo, Shruti Gadewar, Qifan Yang, and Neda Jahanshad. Towards human-guided machine learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 614–624, 2019.

[91] Fabien Girardin and Neal Lathia. When user experience designers partner with data scientists. In *2017 AAAI Spring Symposium Series*, 2017.

[92] Google. People + ai guidebook, 2019.

[93] Google. Visually probe the behavior of trained machine learning models, with minimal coding., 2020.

[94] Robert Grant. Drawmydata a tool for teaching stats and data science, 2020.

[95] Jonathan Grudin. From tool to partner: The evolution of human-computer interaction. *Synthesis Lectures on Human-Centered Interaction*, 10(1):i–183, 2017.

[96] Raymonde Guindon, Herb Krasner, Bill Curtis, et al. Breakdowns and processes during the early activities of software design by professionals. In *Empirical studies of programmers: Second Workshop*, pages 65–82, 1987.

[97] Thilo Hagendorff. The ethics of ai ethics–an evaluation of guidelines. *arXiv preprint arXiv:1903.03425*, 2019.

[98] Lise Amy Hansen. Full-body movement as material for interaction design. *Digital Creativity*, 22(4):247–262, 2011.

[99] Gunnar Harboe and Elaine M Huang. Real-world affinity diagramming practices: Bridging the paper-digital gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 95–104. ACM, 2015.

[100] Gunnar Harboe, Crysta J Metcalf, Frank Bentley, Joe Tullio, Noel Massey, and Guy Romano. Ambient social tv: drawing people into a shared experience. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10. ACM, 2008.

[101] Gunnar Harboe, Jonas Minke, Ioana Ilea, and Elaine M. Huang. Computer support for collaborative data analysis: Augmenting paper affinity diagrams. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1179–1182, New York, NY, USA, 2012. ACM.

[102] Chris Harrison, John Horstman, Gary Hsieh, and Scott Hudson. Unlocking the expressivity of point lights. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1683–1692. ACM, 2012.

[103] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[104] Björn Hartmann, Scott R Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 299–308, 2006.

[105] Rex Hartson and Pardha S Pyla. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.

[106] Mark Hartswood and Rob Procter. Design guidelines for dealing with breakdowns and repairs in collaborative work settings. *International Journal of Human-Computer Studies*, 53(1):91–120, 2000.

[107] Jeffrey Heer. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences*, 116(6):1844–1850, 2019.

[108] Michael Heidt, Andreas Bischof, and Paul Rosenthal. Deconstructivist design within hci. In *International Conference of Design, User Experience, and Usability*, pages 115–122. Springer, 2014.

[109] Joseph M Hellerstein, Vikram Sreekanti, Joseph E Gonzalez, James Dalton, Akon Dey, Sreyashi Nag, Krishna Ramachandran, Sudhanshu Arora, Arka Bhattacharyya, Shirshanka Das, et al. Ground: A data context service. In *CIDR*, 2017.

[110] Karey Helms. Leaky objects: Implicit information, unintentional communication. In *Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems*, pages 182–186. ACM, 2017.

[111] Karey Helms, Barry Brown, Magnus Sahlgren, and Airi Lampinen. Design methods to investigate user experiences of artificial intelligence. In *2018 AAAI Spring Symposium Series*, 2018.

[112] Michi Henning. Api design matters. *Queue*, 5(4):24–36, 2007.

[113] Alex Hern. Twitter apologises for 'racist' image-cropping algorithm, 2020.

[114] Charles Hill, Rachel Bellamy, Thomas Erickson, and Margaret Burnett. Trials and tribulations of developers of intelligent systems: A field study. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 162–170. IEEE, 2016.

[115] Akimitsu Hirota, Masaaki Takemura, and Manabu Mizuno. Design prototyping in" fuzzy front end" of product development-rapid prototyping at the stage of high uncertainty. In *ISPIM Innovation Symposium*, pages 1–14. The International Society for Professional Innovation Management (ISPIM), 2017.

[116] Paul Hodel. The javascript spreadsheet, 2020.

[117] Lars Erik Holmquist. Intelligence on tap: artificial intelligence as a new design material. *interactions*, 24(4):28–33, 2017.

[118] Kenneth Holstein, Erik Harpstead, Rebecca Gulotta, and Jodi Forlizzi. Replay enactments: Exploring possible futures through historical data. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 1607–1618, 2020.

[119] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2019.

[120] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166, 1999.

[121] Daneil Hough. Interfake: Quick json apis, 2020.

[122] Elaine M Huang, Gunnar Harboe, Joe Tullio, Ashley Novak, Noel Massey, Crysta J Metcalf, and Guy Romano. Of social television comes home: a field study of communication choices and practices in tv-based text and voice chat. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 585–594. ACM, 2009.

[123] Elaine M Huang and Khai N Truong. Breaking the disposable technology paradigm: opportunities for sustainable interaction design for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 323–332. ACM, 2008.

[124] Yuner Huang and Ben Young. The art of coupon tests. *Journal of Constructional Steel Research*, 96:159–175, 2014.

[125] Thomas L Huber, Maike AE Winkler, Jens Dibbern, and Carol V Brown. The use of prototypes to bridge knowledge boundaries in agile software development. *Information systems journal*, 2019.

[126] Thomas L Huber, Maike AE Winkler, Jens Dibbern, and Carol V Brown. The use of prototypes to bridge knowledge boundaries in agile software development. *Information systems journal*, 30(2):270–294, 2020.

[127] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 560–575, 2021.

[128] Apple Inc. Designing the ui and user experience of a machine learning app, 2019.

[129] Petra Isenberg and Danyel Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. In *Computer Graphics Forum*, volume 28, pages 1031–1038. Wiley Online Library, 2009.

[130] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241. ACM, 1997.

[131] Robert JK Jacob, Hiroshi Ishii, Gian Pangaro, and James Patten. A tangible interface for organizing information using a grid. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 339–346. ACM, 2002.

[132] Seokhee Jeon, Jane Hwang, Gerard J Kim, and Mark Billinghurst. Interaction techniques in large display environments using hand-held devices. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 100–103. ACM, 2006.

[133] Yuhua Jin, Isabel Qamar, Michael Wessely, and Stefanie Mueller. Photo-chromeleon: Reprogrammable multi-color textures using photochromic dyes. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Emerging Technologies*, pages 1–2, 2020.

[134] Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of ai ethics guidelines. *Nature Machine Intelligence*, 1(9):389–399, 2019.

[135] Tero Jokela and Andrés Lucero. A comparative evaluation of touch-based methods to bind mobile devices for collaborative interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3355–3364. ACM, 2013.

[136] William P Jones and Susan T Dumais. The spatial metaphor for user interfaces: experimental tests of reference by location versus name. *ACM Transactions on Information Systems (TOIS)*, 4(1):42–63, 1986.

[137] Heekyoung Jung and Erik Stolterman. Digital form and materiality: propositions for a new approach to interaction design research. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pages 645–654, 2012.

[138] Elvin Karana, Bahareh Barati, Valentina Rognoli, Anouk Zeeuw Van Der Laan, et al. Material driven design (mdd): A method to design for material experiences. 2015.

[139] Claire Kayacik, Sherol Chen, Signe Noerly, Jess Holbrook, Adam Roberts, and Douglas Eck. Identifying the intersections: User experience+ research scientist collaboration in a generative machine learning interface. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, page CS09. ACM, 2019.

[140] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[141] Zack Kertcher and Erica Coslor. Boundary objects and the technical culture divide: successful practices for voluntary innovation teams crossing scientific and professional fields. *Journal of Management Inquiry*, page 1056492618783875, 2018.

[142] Scott Klemmer, Mark W Newman, and Raecine Sapien. The designer's outpost: a task-centered tangible interface for web site information design. In *CHI'00 extended abstracts on Human factors in computing systems*, pages 333–334. ACM, 2000.

[143] Scott R Klemmer, Anoop K Sinha, Jack Chen, James A Landay, Nadeem Aboobaker, and Annie Wang. Suede: a wizard of oz prototyping tool for speech user interfaces. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 1–10, 2000.

[144] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. Will you accept an imperfect ai?: Exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 411. ACM, 2019.

[145] Michal Kosinski. Facial recognition technology can expose political orientation from naturalistic facial images. *Scientific reports*, 11(1):1–7, 2021.

[146] Peter Kun, Ingrid Mulder, Amalia De Götzen, and Gerd Kortuem. Creative data work in the design process. In *Proceedings of the 2019 on Creativity and Cognition*, pages 346–358. ACM, 2019.

[147] Peter Kun, Ingrid Mulder, and Gerd Kortuem. Design enquiry through data: appropriating a data science workflow for the design process. In *Proceedings of the 32nd International BCS Human Computer Interaction Conference*, page 32. BCS Learning & Development Ltd., 2018.

[148] Hanna Landin. Fragile and magical: Materiality of computational technology as design material. In *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility*, pages 117–120, 2005.

[149] Beth M Lange, Mark A Jones, and James L Meyers. Insight lab: an immersive team environment linking paper, displays, and data. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 550–557. ACM Press/Addison-Wesley Publishing Co., 1998.

[150] Charlotte P Lee. Between chaos and routine: Boundary negotiating artifacts in collaboration. In *ECSCW 2005*, pages 387–406. Springer, 2005.

[151] Charlotte P Lee. Boundary negotiating artifacts: Unbinding the routine of boundary objects and embracing chaos in collaborative work. *Computer Supported Cooperative Work (CSCW)*, 16(3):307–339, 2007.

[152] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer Graphics Forum*, volume 31, pages 1155–1164. Wiley Online Library, 2012.

[153] Susan Leigh Star. This is not a boundary object: Reflections on the origin of a concept. *Science, Technology, & Human Values*, 35(5):601–617, 2010.

[154] Germán Leiva, Nolwenn Maudet, Wendy Mackay, and Michel Beaudouin-Lafon. Enact: Reducing designer–developer breakdowns when prototyping custom interactions. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 26(3):19, 2019.

[155] Nancy G Leveson. Intent specifications: An approach to building human-centered specifications. *IEEE Transactions on software engineering*, 26(1):15–35, 2000.

[156] Yang Li, Jason I Hong, and James A Landay. Topiary: a tool for prototyping location-enhanced applications. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 217–226, 2004.

[157] Rikard Lindell. Code as design material. In *Participatory Materialities Workshop and Symposium at Aarhus University*, 2012.

[158] Zhicheng Liu, Bernard Kerr, Mira Dontcheva, Justin Grover, Matthew Hoffman, and Alan Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. In *Computer Graphics Forum*, volume 36, pages 527–538. Wiley Online Library, 2017.

[159] Natasha Lomas. Twitter may let users choose how to crop image previews after bias scrutiny, 2020.

[160] Arnold M Lund. Measuring usability with the use questionnaire12.". *Usability interface*, 8(2):3–6, 2001.

[161] Lucy Ellen Lwakatare, Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In *International Conference on Agile Software Development*, pages 227–243. Springer, 2019.

[162] Allan MacLean, Richard M Young, Victoria ME Bellotti, and Thomas P Moran. Questions, options, and criteria: Elements of design space analysis. *Human–computer interaction*, 6(3-4):201–250, 1991.

[163] Martin Maguire and Nigel Bevan. User requirements analysis. In *IFIP World Computer Congress, TC 13*, pages 133–148. Springer, 2002.

[164] Ann Majchrzak, Philip HB More, and Samer Faraj. Transcending knowledge differences in cross-functional teams. *Organization Science*, 23(4):951–970, 2012.

[165] Thomas W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, January 1983.

[166] Nirav Malsattar, Tomo Kihara, and Elisa Giaccardi. Designing and prototyping from the perspective of ai in the wild. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, pages 1083–1088. ACM, 2019.

[167] Christopher Manning. Artificial intelligence definitions, 2020.

[168] Miro Mannino and Azza Abouzied. Is this real? generating synthetic data that looks real. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 549–561, 2019.

[169] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilović. How data scientistswork together with domain experts in scientific collaborations: To find the right answer or to ask the right question? *Proceedings of the ACM on Human-Computer Interaction*, 3(GROUP):1–23, 2019.

[170] Tuuli Mattelmäki et al. *Design probes*. Aalto University, 2006.

[171] David Maulsby, Saul Greenberg, and Richard Mander. Prototyping an intelligent agent through wizard of oz. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 277–284, 1993.

[172] Michael McCurdy, Christopher Connors, Guy Pyrzak, Bob Kanefsky, and Alonso Vera. Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1233–1242. ACM, 2006.

[173] Juan Mellado. Aruco javascript, 2018.

[174] Microsoft. Start designing and prototyping, 2020.

[175] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[176] Runway ML. Runway ml: Machine learning for creators, 2018.

[177] Shakir Mohamed, Marie-Therese Png, and William Isaac. Decolonial ai: Decolonial theory as sociotechnical foresight in artificial intelligence. *Philosophy &amp; Technology*, 33(4):659–684, 2020.

[178] Thomas P Moran, Eric Saund, William Van Melle, Anuj U Gujar, Kenneth P Fishkin, and Beverly L Harrison. Design and technology for collaborage: collaborative collages of information on physical walls. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 197–206. ACM, 1999.

[179] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q Vera Liao, Casey Dugan, and Thomas Erickson. How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–15, 2019.

[180] Lauren Murphy, Mary Beth Kery, Oluwatosin Alliyu, Andrew Macvean, and Brad A Myers. Api designers in the field: Design practices and challenges for creating usable apis. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 249–258. IEEE, 2018.

[181] Brad A Myers and William Buxton. Creating highly-interactive and graphical user interfaces by demonstration. *ACM SIGGRAPH Computer Graphics*, 20(4):249–258, 1986.

[182] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. Protoar: Rapid physical-digital prototyping of mobile augmented reality applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[183] Donald A Norman and Stephen W Draper. *User centered system design: New perspectives on human-computer interaction*. CRC Press, 1986.

[184] William Odom and Tijs Duel. On the design of olo radio: Investigating metadata as a design material. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–9, 2018.

[185] Gregory B Olson. Designing a new material world. *Science*, 288(5468):993–998, 2000.

[186] Oracle. Powerful api design stack, 2020.

[187] John Ousterhout. *A Philosophy of Software Design*. Yaknyam Press, 2018.

[188] Bora Pajo. Food choices: College students' food and cooking preferences. `hhttps://www.kaggle.com/borapajo/food-choices`, 2017.

[189] David Lorge Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.

[190] Kayur Patel. Lowering the barrier to applying machine learning. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 355–358, 2010.

[191] Dorian Peters, Karina Vold, Diana Robinson, and Rafael A Calvo. Responsible ai—two frameworks for ethical design practice. *IEEE Transactions on Technology and Society*, 1(1):34–47, 2020.

[192] Nadia Piet. Ai meets design, 2019.

[193] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4, 2005.

[194] Peter G Polson, Clayton Lewis, John Rieman, and Cathleen Wharton. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, 36(5):741–773, 1992.

[195] John Pruitt and Jonathan Grudin. Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences*, pages 1–15, 2003.

[196] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019.

[197] Lavanya Ramakrishnan, Sarah Poon, Valerie Hendrix, Daniel Gunter, Gilberto Z Pastorello, and Deborah Agarwal. Experiences with user-centered design for the tigres workflow api. In *2014 IEEE 10th International Conference on e-Science*, volume 1, pages 290–297. IEEE, 2014.

[198] Johan Redström. On technology as material in design. *Design Philosophy Papers*, 3(2):39–54, 2005.

[199] Yim Register and Amy J Ko. Learning machine learning with personal data helps stakeholders ground advocacy arguments in model mechanics. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*, pages 67–78, 2020.

[200] Holger Rhinow, Eva Köppen, and Christoph Meinel. Prototypes as boundary objects in innovation processes. In *Proceedings of the 2012 International Conference on Design Research Society, Bangkok, Thailand*, 2012.

[201] Mark O Riedl. Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies*, 1(1):33–36, 2019.

[202] Erica Robles and Mikael Wiberg. Texturing the” material turn” in interaction design. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 137–144, 2010.

[203] Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.

[204] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

[205] Tobias Schnabel, Paul N Bennett, and Thorsten Joachims. Improving recommender systems beyond the algorithm. *arXiv preprint arXiv:1802.07578*, 2018.

[206] Donald Schon and John Bennett. Reflective conversation with materials in bringing design to software, winograd t, 1996.

[207] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511, 2015.

[208] Raymond Scupin. The kj method: A technique for analyzing data derived from japanese ethnology. *Human organization*, pages 233–237, 1997.

[209] Ahmed Seffah, Jan Gulliksen, and Michel C Desmarais. *Human-centered software engineering-integrating usability in the software development lifecycle*, volume 8. Springer Science & Business Media, 2005.

[210] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[211] Ben Shneiderman. Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered ai systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 10(4):1–31, 2020.

[212] Ben Shneiderman. Human-centered artificial intelligence: Reliable, safe &amp; trustworthy. *International Journal of Human–Computer Interaction*, 36(6):495–504, 2020.

[213] HA Simon. The sciences of the artificial (1996; orig. ed. 1969; 2nd, rev), 1969.

[214] Nishant Sinha and Rezwana Karim. Responsive designs in a snap. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 544–554, 2015.

[215] David Canfield Smith. Pygmalion: a creative programming environment. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1975.

[216] Jared M. Spool. Apis: The future is now, 2013.

[217] Robert Stalnaker. Common ground. *Linguistics and philosophy*, 25(5/6):701–721, 2002.

[218] Susan Leigh Star. The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving. In *Distributed artificial intelligence*, pages 37–54. Elsevier, 1989.

[219] Luke Stark. Facial recognition is the plutonium of ai. *XRDS: Crossroads, The ACM Magazine for Students*, 25(3):50–55, 2019.

[220] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.

[221] Drew Steedly, Chris Pal, and Richard Szeliski. Efficiently registering video into panoramic mosaics. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1300–1307, Washington, DC, USA, 2005. IEEE Computer Society.

[222] Hariharan Subramonyam, Steven M Drucker, and Eytan Adar. Affinity lens: Data-assisted affinity diagramming with augmented reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[223] Hariharan Subramonyam, Wilmot Li, Eytan Adar, and Mira Dontcheva. Taketoons: Script-driven performance animation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 663–674, 2018.

[224] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. Protoai: Model-informed prototyping for ai-powered interfaces. In *26th International Conference on Intelligent User Interfaces*, pages 48–58, 2021.

[225] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. texsketch: Active diagramming through pen-and-ink annotations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[226] Petra Sundström, Alex S Taylor, and Kenton O'Hara. Sketching in software and hardware bluetooth as a design material. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 405–414, 2011.

[227] Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*, 2019.

[228] Ivan E Sutherland. Sketchpad a man-machine graphical communication system. *Simulation*, 2(5):R–3, 1964.

[229] Kendra T. Airbnb – using ai to evaluate if a guest is trustworthy, 2020.

[230] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018.

[231] Sandeep Tata, Alexandrin Popescul, Marc Najork, Mike Colagrosso, Julian Gibbons, Alan Green, Alexandre Mah, Michael Smith, Divanshu Garg, Cayden Meyer, et al. Quick access: building a smart experience for google drive. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1643–1651. ACM, 2017.

[232] Edward Tse, Saul Greenberg, Chia Shen, Clifton Forlines, and Ryo Kodama. Exploring true multi-user multimodal interaction over a digital table. In *Proceedings of the 7th ACM conference on Designing interactive systems*, pages 109–118. ACM, 2008.

[233] Blase Ur, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin, Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini, Hana Habib, et al. Design and evaluation of a data-driven password meter. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3775–3786, 2017.

[234] Anna Vallgårda and Johan Redström. Computational composites. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 513–522, 2007.

[235] Philip van Allen. Prototyping ways of prototyping ai. *interactions*, 25(6):46–51, 2018.

[236] Willemien Visser. *The cognitive artifacts of designing*. CRC Press.

[237] Eric Von Hippel. “sticky information” and the locus of problem solving: implications for innovation. *Management science*, 40(4):429–439, 1994.

[238] Jagoda Walny, Christian Frisson, Mieka West, Doris Kosminsky, Søren Knudsen, Sheelagh Carpendale, and Wesley Willett. Data changes everything: Challenges and opportunities in data visualization design handoff. *IEEE transactions on visualization and computer graphics*, 26(1):12–22, 2019.

[239] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. Designing theory-driven user-centric explainable ai. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2019.

[240] Mikael Wiberg. Interaction, new materials &amp; computing–beyond the disappearing computer, towards material interactions. *Materials &amp; design*, 90:1200–1206, 2016.

[241] William Widjaja, Keito Yoshii, Kiyokazu Haga, and Makoto Takahashi. Discusys: Multiple user real-time digital sticky-note affinity-diagram brainstorming system. *Procedia Computer Science*, 22:113–122, 2013.

[242] James Wilson and Daniel Rosenberg. Rapid prototyping for user interface design. In *Handbook of human-computer interaction*, pages 859–875. Elsevier, 1988.

[243] Maike AE Winkler, Carol Brown, and Thomas L Huber. Recurrent knowledge boundaries in outsourced software projects: A longitudinal study. In *ECIS*, 2015.

[244] Terry Winograd, Fernando Flores, and Fernando F Flores. *Understanding computers and cognition: A new foundation for design*. Intellect Books, 1986.

[245] Wireframe—CC. A design tool fine tuned for wireframing, 2020.

[246] Larry E Wood. *User interface design: Bridging the gap from user requirements to design*. CRC Press, 1997.

[247] William Wright, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. The sandbox for analysis: concepts and methods. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 801–810. ACM, 2006.

[248] Wei Xu. Toward human-centered ai: a perspective from human-computer interaction. *Interactions*, 26(4):42–46, 2019.

[249] Qian Yang. The role of design in creating machine-learning-enhanced user experience. In *2017 AAAI Spring Symposium Series*, 2017.

[250] Qian Yang. Machine learning as a ux design material: How can we imagine beyond automation, recommenders, and reminders? In *2018 AAAI Spring Symposium Series*, 2018.

[251] Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T Iqbal, and Jaime Teevan. Sketching nlp: A case study of exploring the right things to design with language intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 185. ACM, 2019.

[252] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. Investigating how experienced ux designers effectively work with machine learning. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 585–596. ACM, 2018.

[253] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pages 1–13, 2020.

[254] Zhibin Zhou, Lingyun Sun, Yuyang Zhang, Xuanhui Liu, and Qing Gong. Ml lifecycle canvas: Designing machine learning-empowered ux with material lifecycle thinking. *Human–Computer Interaction*, pages 1–25, 2020.