Form to Flow: Exploring Challenges and Roles of Conversational UX Designers in Real-world, Multi-channel Service Environments

JEONGYUN HEO, Kookmin University, Republic of Korea and KAIST, Republic of Korea
UICHIN LEE*, KAIST, Republic of Korea

Conversational agents are widely used in today's multi-channel service environments. However, little is known regarding the challenges faced by user experience (UX) designers. This study explores the challenges faced by conversational UX designers working in multidisciplinary teams for understanding the design process involved in the transformation of conventional graphical interfaces to conversation flows. In-depth interviews with UX designers working in industries reveal the key challenges in the form-to-flow transformation process. Moreover, collaborative work with various stakeholders in complex work environments involves conversational artificial intelligence-engendered gap-filling work phenomena wherein UX designers tend to work across role boundaries. Our results indicate the need for added support for CUX designer including tools and guidelines to suppport form-to-flow design, tools for testing, and defining extended roles for CUX designers with collaborative perspectives for designing conversational agents in multi-channel service environments.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**.

Additional Key Words and Phrases: Conversational User Experience Design, Collaborative Design Process

ACM Reference Format:
Jeongyun Heo and Uichin Lee. 2023. Form to Flow: Exploring Challenges and Roles of Conversational UX Designers in Real-world, Multi-channel Service Environments. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW2, Article 340 (October 2023), 24 pages. https://doi.org/10.1145/3610189

1 INTRODUCTION

With the adoption of advanced natural language processing (NLP) technology based on machine learning (ML) and artificial intelligence (AI), chatbots are currently being used to enable various services, such as personal assistants, smart home control, e-commerce, banking, and healthcare support [12, 36, 57]. While chatbots are often referred to as text-based chatbots, conversational AI, and digital companions, this work uses the term conversational agent (CA) as it embodies various terminologies. A CA can be described as "software that receives natural language as input and generates natural language as output, thus conversing with the user" [25].

The design process for CA user experience (CUX) includes the design of a conversation flow and pattern between a human and a system based on natural language [16, 46, 59]. Considering that most designers are trained in graphical user interfaces (GUI), object-based web platforms, and app-based environments, and current human-computer interaction (HCI) training is organized accordingly [49, 50], several user experience (UX) designers are unfamiliar with CUX design. This

*The corresponding author

Authors' addresses: Jeongyun Heo, yuniheo@kookmin.ac.kr, Kookmin University, Seoul, Republic of Korea and KAIST, Daejeon, Republic of Korea; Uichin Lee, uclee@kaist.ac.kr, KAIST, Daejeon, Republic of Korea.

2573-0142/2023/10-ART340 $15.00
https://doi.org/10.1145/3610189

observation is based on the fact that CUX design requires the knowledge of several domains, such as conversational flow design, in addition to technology, such as ML and NLP, which are departures from traditional software development [1, 2, 58]. For new CUX designers who have undergone training in a conventional UX design setting, the requirement of knowledge and comprehension of technologies such as ML and NLP may act as barriers.

Due to the diversity of user interaction modalities, various online services are often provided over *multiple channels* such as the web, mobile apps, messenger platforms, chatbots, and smart speakers. A channel is a specialized platform or medium through which a service provider interacts with its customers. Service providers often offer multiple channels to users in an effort to maximize user contact. Further, it is important to deliver a consistent and seamless experience over multiple channels. Because CAs offer natural language interaction, service providers often consider CA support to be essential for increasing service accessibility. According to Jupiter, which is a market trend research institution, chatbots are one of the key channels in this multi-channel expansion strategy, and it is anticipated that by 2023, 50% of chatbots will supplement or replace mobile application functions [60]. CUX design may be categorized as either a standalone CUX service that operates as an independent channel or as a multi-channel service where a CA is used along with other interaction channels such as the web and mobile apps. In this work, we focus on the design of CUX services for multi-channel service environments.

The goal of this study was to deepen our understanding of the CUX design process, which has expanded significantly with the development of CA services in multi-channel service delivery environments. CUX designers often start experiencing the design process using commercially developed dialogue management systems (DMS) and chatbot builders with their mental models already established based on mobile app and web design. The majority of designers in the domain of CUX are accustomed to GUI-style design. In general, their metaphor for the GUI-style design mental model is 'form', a collection of visible GUI objects, whereas, in the case of CUX, a 'flow' metaphor is used to explain the turn-by-turn conversation dialogue process. Developing CUX design for expanding multi-channel service delivery can be considered as a form-to-flow transition. This new design process motivates us to answer the following research questions (RQs): RQ1: What are the work strategies of a CUX designer for multi-channel service design, and what are the main barriers faced by designers engaged in CUX design? RQ2: What are the key difficulties confronting CUX designers in multi-channel service environments? RQ3: How do the roles of CUX designers differ from those of conventional GUI-based designers, and how does collaborative work with different stakeholder groups evolve through the course of the CUX design process?

We interviewed 17 designers who had experience in both UX and CUX design to investigate the design processes, methodologies, and practices used by CUX designers working with interdisciplinary teams in multi-channel service environments. We then summarized the key challenges experienced by CUX designers and their underlying causes. Our findings provided useful insights into building new design tools and collaboration aids. This study is the first step toward a comprehensive understanding of the challenges faced by CUX designers that future studies can consider.

2 BACKGROUND AND RELATED WORK

This section provides an overview of prior studies related to CA design and development. Contrary to a typical GUI-centered design process, this study focused on the investigation of the differences in the roles expected to be played by CUX designers and the differences in the CA design processes.

2.1 CA as an Interaction Modality Extension

The advent of platform-based CAs, such as Amazon Echo and Google Assistant, has resulted in the exponential growth of CAs and pressure on service providers to reach customers via multiple

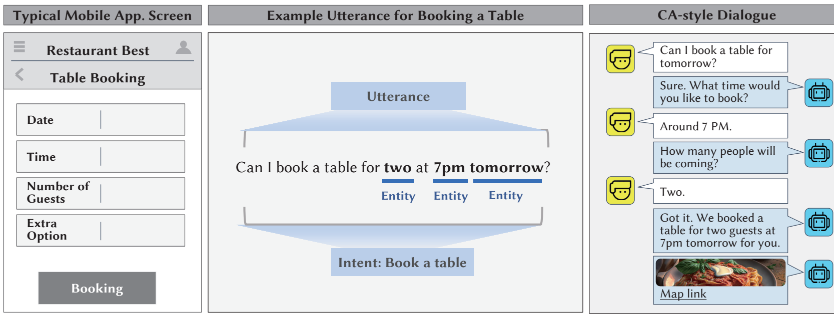

Fig. 1. Illustrative example of intent and entities

channels [3, 36, 45, 45]. For example, a chatbot can be used to purchase an airline ticket; however, this can also be done through an existing website, mobile application, or offline kiosk. In particular, there is a rapid growth in the adoption of CAs in sectors where the availability of multiple channels that can be accessed 24h a day is related to sales in various industries, including the tourist, retail, food and beverage, and banking industries [15, 54, 63]. In general, the customer contact points for service delivery may vary based on the modality of the interaction and device use. For effective service operation, all channels should provide services over a single data management backend instead of separate treatments for individual channels. Chatbot services are widely used for channel expansion of existing multi-channel service environments wherein other channels, such as mobile and web-based applications, are in operation for delivering this service. The work environment of UX designers may be categorized based on the size of the organization, the size of the team, the roles of team members, and the size of the development service. The major characteristics of CA services are developed for channel expansion in environments where service channels are in operation.

2.2 Design Pattern of Conversational Agents

Conversational agents can be classified into task-oriented services such as airline ticket bookings and restaurant booking, and information-oriented services, which include frequently asked question (FAQ)-oriented queries and answers. This study mainly focused on the design of task-oriented CAs. For current users accustomed to performing various tasks digitally, the most powerful task model is a mobile application [65]. For this reason, Li and Riva [43] suggested a method that automatically interprets the GUI of mobile apps to extract user tasks and generate questions and answers for CAs utilizing the most prevalent slot-filling strategy for developing CAs. Arsan et al. [6] developed a service-similar to Google Assistant and Siri-that completes the input requested by the user to capture a snapshot of the GUI screen, thus targeting CA services that are used in conjunction with mobile phones. Although the task model can be automatically extracted from the mobile application, it is generally perceived that the CA experience design requires significant additional user research. Furthermore, most designers are trained in GUI object-, web-, and app-based contexts [51], however, CUX design often requires an understanding of new technologies such as NLP and ML and key CUX design building concepts such as intent and entity that would be challenging for new CUX designers. The intent of a given request represents a user's underlying goal or objective, whereas an entity (or slot) refers to the required information to achieve that objective, as illustrated in Figure 1.

GUI objects such as radio buttons, checkboxes, scrolls, and hamburger menus are extensively employed as design patterns. Mouse-based interaction methods such as drag and drop, click, double click, and touch-based pinch user interfaces (UIs), which allow for intuitive magnification/demagnification using two fingers, are the result of decades of design work and market acceptance [14, 27, 61]. When a new technology, such as a touch screen, is launched, the intent of the designer is to provide intuitive operation to the user, and the effort directed toward its implementation is integrated, thus resulting in a standardized UI that designers, developers, and users implicitly agree upon. Ozenc et al. [55] defined the "immaterial" materiality of software based on observations from designers developing new controls. Nelson and Stolterma first used the term "ultimate particular" to refer to a standardized UI element, by which a new concept from the designer, which was related to the operation of a device was established via a process of tacit consensus, implementation, and iterative improvement [52]. "Ultimate particular" emphasizes the significance of concrete design solutions as expressions of knowledge.

A design pattern is a standardized method of description used to specify the solution to a problem systematically. Since its initial use by the architect C. Alexander, it has been utilized in other domains, including architecture, computer science, and HCI [4, 11, 62]. Kruschitz and Hitz [37] focused on knowledge transfer using design patterns in the field of HCI. A design pattern may be regarded as an "ultimate particular," as mentioned above, and most designs are built on design patterns in *stable development environments* such as applications and web-based platforms.

In newly introduced design domains such as CAs, a consensus among design patterns is yet to be established. Thus, the connection between a design concept and the actual implementation process is tenuous. It is evident that the designer is required to communicate the delivery of a new experience to the developer. Designers are expected to translate their ideas into interactions that can then be implemented in conjunction with the developers. User experience designers who are *unfamiliar with a technology* are often uncertain about the feasibility of a development; developers, especially, are unfamiliar with the development environment. The process of reaching a consensus can be considered to be a sequence of challenges [17, 21, 55]. In addition, establishing a suitable criterion for choosing between feasibility of an environment with limited development resources and acceptability in the current (evolving) development environment is challenging. In this context, we derived the following research questions: RQ1: What are the work strategies of a CUX designer for multi-channel service design and what are the main barriers faced by designers engaged in CUX design? RQ2: What are the key difficulties confronting CUX designers in multi-channel service environments?

2.3 Collaborative Work within a Multidisciplinary Team

Designers collaborate within a multidisciplinary team using AI/ML and NLP technologies as design elements. The NLP model, which is one of the technologies at the core of CA implementation, is representative of AI technologies based on ML. Due to the probabilistic nature of AI models and the difficulties associated with understanding their technicalities, the design process employing AI models presents several obstacles to UX designers [28, 64, 66, 67, 69, 70, 72]. Zdanowska and Taylor [72] noted that, although numerous academic studies were conducted in the field of AI/ML with respect to HCI, there was minimal access to the development process of these systems in the "real world." The role of current UX practitioners was evaluated with an emphasis on corporate ML services in the business-to-business domain. Yang et al. [70] identified the difficulties associated with the human-AI interaction design process due to the *uncertainty* of AI capabilities and the *complexity* of AI models and proposed a map of the complexities involved in AI design. Considering the challenges faced by AI UX, especially regarding the difficulties associated with AI failure prediction due to the probabilistic nature of AI models and the complexities involved in testing prior to system deployment, Hong et al. [29] presented a low-cost prototyping tool for ML services.

Several researchers noted that AI/ML systems were changing the traditional user-centered design approach. Yang et al. [69] focused on a method involving collaboration between designers and data scientists in a new technology-oriented collaborative environment. Piorkowski et al. [56] analyzed how AI developers overcame communication gaps within a multidisciplinary team with respect to the lack of a shared mental model and provided information to stakeholders who collaborated using documentation tools such as Microsoft PowerPoint, despite the high cost involved. Thus, the third research question was derived within this context as follows, "RQ3: How do the roles of CUX designers differ from those of conventional GUI-based designers, and how does collaborative work involving different stakeholder groups evolve during the CUX design process?" This includes the challenges faced during collaborative work.

3 METHODOLOGY

The problems faced by CUX designers in industrial sites were collected and analyzed using graphic elicitation-based in-depth interviews.

3.1 Participant Recruitment

The selection criteria for the interview participants were UX designers with UI/GUI design experience and task-oriented CA development experience. A total of 17 designers (12 females and 5 males) were recruited from three companies. Three of them were removed from the data analysis process because their participation in the interview was insincere, the project type was not in the multichannel environment, or the role of participation was auxiliary. Each interview participant was assigned a code for identification (refer to the summary of participants in Table 1). This study was approved by the institutional review board (IRB). Pseudonyms were assigned and managed to protect the personal information of those interviewed. Informed consent was obtained after the objectives of the study were clarified.

3.2 Interview Details and Analysis Methods

The interviews were conducted for approximately 60 to 90 min via video conferencing (Zoom and Google Meet), and those interviewed received a gift certificate worth 50,000 KRW (approximately 40 USD). The interview process was recorded with informed consent, and the data was collected in the form of audio recordings, notes, and drawings. For the purpose of explanation, the interviewee drew or added notes directly on the paper. The drawings were captured using a cell phone and sent online to the interviewer.

The in-depth semi-structured interview questions were composed based on the followings: 1) the current work environment, including the chatbot development process and preferred tools, 2) the role played by the designer and the issues faced during chatbot design, 3) differences between GUI/UI and conversational user interface (CUI), 4) the designer's level of understanding of CUX design concepts such as intent and entity design, and 5) a simple task was assigned to each participant to extract the designer's knowledge of the subject. Graphic elicitation techniques and in-depth interviews [8] were employed to better interpret the thoughts and experiences of the designers. In particular, we used the illustrations sketched by the designer and freely recorded notes to verbally discuss the important aspects. The knowledge extraction approach that incorporates visual expression facilitated the expression of empirical and tacit knowledge, which is challenging to express verbally [23]. These visual representations were reported to have enhanced the ability of those interviewed to express exclusive thoughts [9] and the in-depth interviews and communication between the interviewers and subjects [20]. The list of tasks created to extract the knowledge of the designer was as follows: 1) a comparison of existing application-/web-based development processes

Table 1. Overview of Participants. Conventional UX and CUX design experiences in years. Design experiences include major products and multi-channel lists.

No	Code	Gender	UX Exp.	CUX Exp.	Company	Design Experiences
1	D1	M	13	5	Major IT company for service development	UX for a smart speaker platform Web, App, CA
2	D2	F	12	6		Chatbot for smart home services Customer chatbot for e-commerce Web, App, CA
3	D3	M	11	4		Smart speaker for smart homes Vehicle infotainment CA
4	D4	F	10	7		Customer chatbot for insurance services Web, App, CA
5	D5	F	6	6		AI agent for in-vehicle services Customer chatbot for credit card services Web, App, CA
6	D6	F	9	4	Major IT company offering IT integration services	Customer chatbot for banking Web, App, CA
7	D7	F	11	4		Customer chatbot for home shopping Web, App, CA
8	D8	M	9	5		Customer chatbot for banking Web, App, CA
9	D9	F	21	3		Customer chatbot for credit card services Web, App, CA
10	D10	F	5	4		Customer chatbot for credit card services Web, App, CA
11	D11	F	17	7	Major design agency	Customer chatbot for credit card services Web, App, CA
12	D12	F	8	4		Customer chatbot for credit card services Web, App, CA
13	D13	M	13	4		Customer chatbot for credit card services Web, App, CA
14	D14	F	7	2		Customer chatbot for tourist info services Web, App, CA

and CUX development processes, 2) intent design for a simple task, and 3) the clarification of the similarities and differences between existing application/web-based and CUX development items.

After the interview, we transcribed the recorded sessions and analyzed the collected data. We followed the approach proposed by Braun and Clarke to perform the thematic analysis [13] wherein the analysis of collected data consisted of six steps: familiarizing oneself with the data, generating initial codes, searching for themes, reviewing, defining and naming them, and producing the report. We began our analysis with an initial familiarization phase in which the transcripts and drawings were carefully reviewed. Relevant words, phrases, or sentences containing significant expressions were coded (a total of 150 codes were found) to generate the initial themes [26]. By following an iterative process of re-reading and coding, we identified four main themes and 14 sub-themes related to the research questions.

3.3 Participant Characteristics

The average experience in UX design of those interviewed was 10.9 years. The shortest experience was 5 years, while the longest was 21 years. Those interviewed had an average of 4.6 years of experience in conversational UX design, with the least being of 2 years and the highest of 5 years. All the designers had transitioned to CUX after general UX work.

Types of CUX Projects: It was observed that all the participants had worked on CUX service projects in multichannel environments, of which web-based and mobile applications were the most common. There were cases in which hardware-based specialized devices were used in parallel with mobile applications and ones in which the designers mentioned standalone CUX services that were operated in conjunction with a telephone call-based counselor or a web page-based FAQ service; thus, they could not be considered standalone types. The affiliations of the respondents were classified into three categories: (1) a major organization that developed its own services, (2) a large IT company that provided IT integration services, and (3) a large design agency. The expertise of the designers participating in the interviews was identified by their respective professions. The scale of the participating tasks varied depending on the maximum number of intents to be fulfilled by the chatbots (100–500 across firms). Several designers were engaged in development projects for smart speaker platforms and text-based chatbots. It was evident that the participating designers had expertise in designing commercial CA product services.

Team Structure and Size: Two or three CUX designers were assigned to each project, regardless of the size of the company with which they were affiliated. In the case of one firm, a large team with more than ten CUX designers was assigned to develop a new service category, but the number of designers per service module was 1–3.

CA Implementation Platform: Conversational UX development was found to be dependent on various platforms that supported dialogue management and NLP, including Google, Amazon, Naver, and Kakao. Open source-based custom platforms were also employed. A single service was often implemented using multiple platforms (possibly to increase service accessibility).

4 RESULTS

This section provides a detailed illustration of the major themes: a form-to-flow strategy for CA design in multichannel service design (RQ1); a lack of common design patterns and implementation feasibility leading to reworking (RQ2); and the complexity of collaborative work environments lacking shared understanding, and the gap-filling tasks of the designer due to a lack of clarity on the role of CUX designers (RQ3).

4.1 Form-to-Flow: CUX Design Process Differences and CUX Barriers

Based on the interviews, it was observed that despite the rapid expansion of CUX design in the industry, only a few UX designers were educated exclusively in CUX design. Most CUX designers migrated to CUX design after receiving training in a CUX-related area to expand their design field, although their initial skill sets were acquired for mobile applications or web-based platforms. All the cases; in which the CUX service designers participated, involved the extension of CUX services in operations such as web-based or mobile applications to multiple channels. Examples include the application of a chatbot to a web-based tourism information service and an app-based ticket and train ticket reservation service.

Based on the results of the interview, it can be inferred that the experience of the designer and that of the user, was formed based on the mobile application or web-based platform where most services are provided. Moreover, the role of CUX designers was to convert form-type design, such as a web-based or mobile application, into a conversational flow. The interviews revealed the

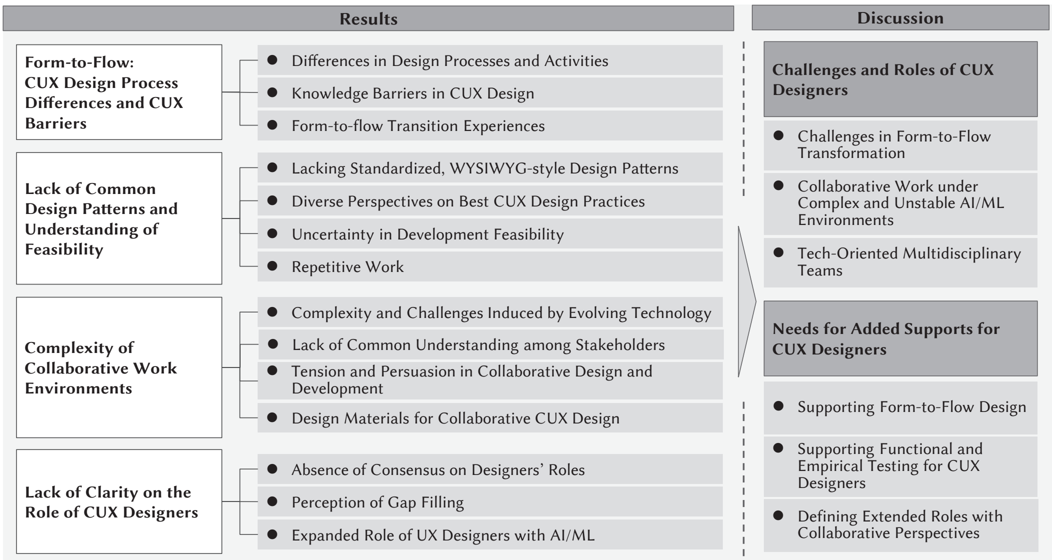

Fig. 2. Results of thematic analysis and topics discussed

following: 1) the differences in activities related to the design process of CUX design compared with those of a mobile application or web platform; 2) the knowledge barriers when engaging in CUX design; 3) major experiences in form-to-flow conversion.

4.1.1 Differences in Design Processes and Activities. Interpretation of the perceptions of the designer using graphic elicitation revealed that the configuration phases of the CUX development process were the same as those of the GUI design process in that they were composed of the following steps: analysis, design, prototyping, development, testing, and product launching. However, the activities in the individual stages differed. Figure 3 summarizes the overall differences. Here, drawings were used to collect the designers' perceptions required to acquire an extensive understanding of the changes in the design process at each of the stages mentioned. As a result, it was found that the features of intent/entity design work and the testing phase were distinctive, and there were several repeated activities, as stated by a majority of the designers who participated in the interview. Additional differences unique to the CUX design process included the generation of training data for DMS, which is the foundation of any machine learning-based system. In general, the process of generating training-domain-specific and task-oriented datasets starts with the generation of representative sample sentences. These sample sentences could be further used to increase the dataset size (e.g., by designers generating the datasets, outsourcing dataset generation, or leveraging customer interaction data such as call center logs). This process was then followed by inputting the dataset into the DMS system for model building. It was observed that CUX designers were predominantly engaged in the sample sentence generation and input phases of the training data generation process for DMS.

Most CUX designers considered the information architecture stage as equivalent to the intent/entity design stage; both are functions of the information design process. The intent/entity design stage may be interpreted as a type of information structure design, which is at the core of the logical design process for CAs. The addition of data collection and learning process tasks were identified as the pre-analysis phases; these are not found in mobile apps or web designs. In

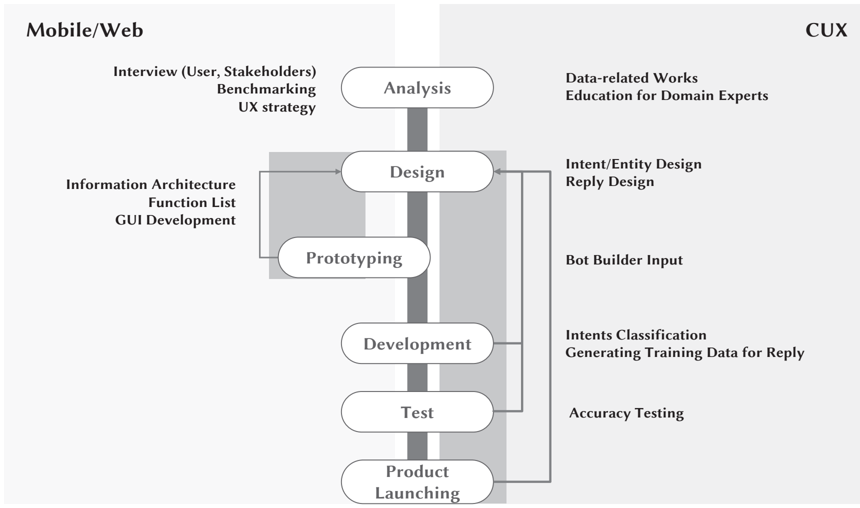

Fig. 3. Differences in the design processes of mobile/web and CUX design

addition, numerous designed outputs were different from what was expected, which resulted in a large amount of reworking, causing designers to return to the previous stage. The absence of an appropriate testing method was highlighted as a significant difference and challenge when compared with mobile applications and web-based design. "*If they (i.e., development team) say no, the intent itself should be modified or changed*" (D5).

The diagram drawn by D10, which is presented in Figure 4 illustrates why such repetitive processes occur and the difficulties faced by CUX designers. The UI design in existing mobile or web applications is very stable, just as at the top of a tower with a solid foundation; however, with CUX, the design elements are hardly visible, even when the visible elements are emphasized. As a result, it is difficult to grasp any of the structure other than the visible elements, as in the case of the "*tip of the iceberg*." Furthermore, it is very inconvenient to use incorrectly designed CUX services.

The most representative case is the explanation of the difference in design activities that occur during the conversion from a form-based design to a conversational flow design, such as intent/entity identification, generation of training sentences, an inspection of NLP input/output processing: understanding the user's text input to extract relevant information, return to the correct and intended output, and response modeling. In several cases, the role was to consider the characteristics of the CUX development environments, as they were not clarified when making the distinction between the processes, for example, the determination of the depth of information and the design of policies for exceptional cases. For example, the emphasis on policy design for exceptions explains the need for additional design activities by highlighting the inability of the current DMS engine to respond as expected. Several designers highlighted output formats, such as standard template management; this clarifies the expanded role of UX designers in contrast to those working on mobile apps and web-based platforms, given that standard design documents are yet to be established in CUX design. The assumption of this role by UX designers leads to clarification on "*discussions for collaborations*" and "*requests for involvement in content-related development*," which may be regarded as requirements for a new role that cannot be clarified based on the current process. There were variations in the labor involved at each stage of a given process, and this explains why there was a

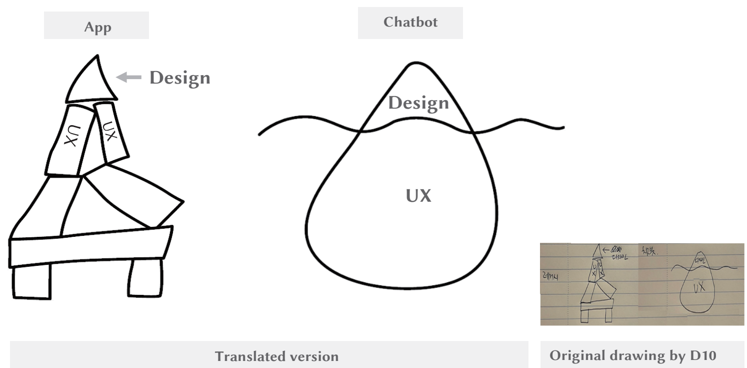

Fig. 4. Illustration by D10 elaborating on the characteristics of CUX design

significant change in the actual work in the overall design process. "*I thought that design was merely a shell in the chatbot field compared to other fields*" (D10). Participant D4 emphasized intent/entity design, flow design, response modeling, and text inspection. The emphasis on policy design for exceptional cases was judged based on the results of experiences where the existing DMS engines could not respond when it was not possible to provide exact answers.

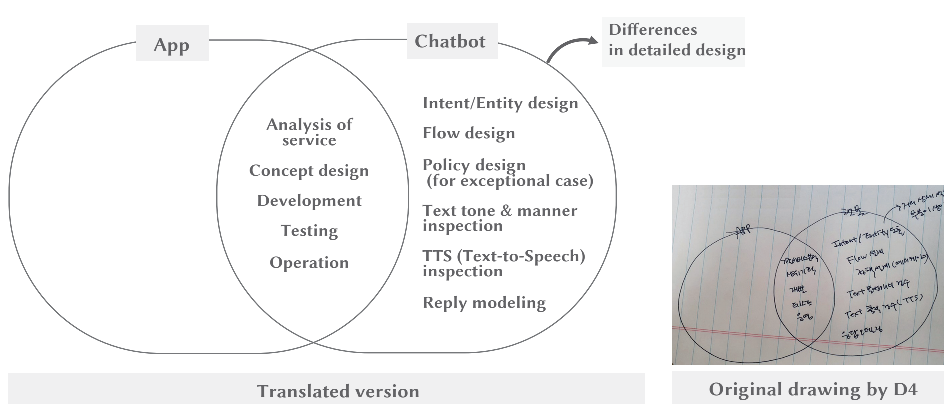

Fig. 5. Illustration by D4 elaborating on the differences in design activities

4.1.2 Knowledge Barriers in CUX Design. For CUX design, it is necessary to acquire knowledge that is different from that required for the current mobile applications/web platforms. The comprehension of intent/entity is most critical, in addition to other DMS characteristics, along with the comprehension of bot builders, conversation models, and NLP capabilities.

(1) Understanding the basic characteristics of conversational UX design: Intent and entity can be identified as the most fundamental knowledge required for CUX design, in addition to being the most significant distinction from form-based design based on UI objects. "*Those who worked on menu-based interfaces are probably unfamiliar with the idea of intent and entity*" (D4). "*Difficulties*

occur when a person designing a chatbot uses traditional UX models to understand the concepts of intent and entity" (D8).

Branch design: The metaphor of a flow is generally used to describe CUX design, which consists of a sequential arrangement of intents, with the subsequent intents determined by the response of the entity in the previous intent stage. Typically, a flowchart is used to illustrate the order of the intents according to these entities. The design of subsequent intents based on the entity response is referred to as a branch design. As expressed by participant D4, "*If I were asked to select one term that appropriately expresses the challenge of conversational design, it would be a branching hell.*" The expression of the "*hidden intent structure*" or "*how to split intent*" expressed by D7 indicates branching design.

(2) Understanding technologies including their limits: Among the CUX designers who participated in the interview, designers who were experienced in tasks involving a large number of intents explained the limitations of technology. Designers mentioned the understanding of natural language unit characteristics using ML. We also found difficulties resulting from excessive expectations related to AI performance, such as cases of additional learning of example sentences required for specific services and the reduction of simple tasks such as learning example sentences. "*If you ask about the expectations that consumers or users have when they refer to AI, they are of the opinion that the system will automatically respond to all queries. Indeed, there are problems that occur because expectations deviate from feasible reality*" (D8). In addition, designers reported their experiences when validating the requirement of designing intents, considering the performance of the engine after actual service implementation, and changing interaction designs due to implementation limitations at the meeting stage for implementation after planning. "*I have noted that the UX is not always as intended due to the issue of actual backend engine performance*" (D8).

Several CUX designers mentioned differences in the platform characteristics, excessive dependence on each platform, and incompatibilities between platforms. In fact, multiple platforms co-existed, and the platform technology was at the center of the design stage as well. "*There are occasions wherein people on different platforms experience difficulties in communicating with each other*" (D6). "*Google Dialogflow updated something. Then we had to work on the previous development again*" (D9).

4.1.3 Form-to-flow Transition Experiences. A mobile application or webpage, which is a collection of visible GUI objects, is a metaphor for 'form.' In the case of CUX, the metaphor for flow was used to explain consecutive queries and responses (turn-by-turn) that occur between the user and the system during the dialogue process. CUX designers who participated in the interview played a role in converting form-based UX, made up of UI components that provided visual affordance, into a conversational flow. "*The initial chatbot was basically content that was transferred from the application screen to the chatbot*" (D14) The majority of the projects carried out by the CUX designers who participated in the interviews were related to multi-channel services that added CUX services to existing GUI-based mobile applications or online services. "*The chatbot project is not solely initiated by the development of chatbots. For example, if a company intends to develop an online channel, the first step is to create a web homepage on a personal computer (PC), followed by a mobile application, and finally, a chatbot. With respect to the overall UX, scenarios should be developed for all aspects connected to and harmonized with the entire service. This is a difficult task*" (D8).

During the form-to-flow transition, the process of restructuring the UX to suit flow-type interaction includes different considerations when compared with the design of the existing form-based UX. In the context of form design, the primary goal is to provide affordance through visual GUI objects on the screen; this allows users to complete specified tasks. The GUIs also constrain user interactions in certain contexts (e.g., making a field or button unmanipulable). Flow design, however, cannot restrict the scope of available intents at any given moment. The role of the designer is to organize a conversational flow by defining preferable tasks as distinct intents and identifying the user input required in each intent as entities. These different considerations involve design exploration, often leading to an iterative and repetitive process. The description of DL1 below confirms that the design of flow is recognized as a distinct function of CUX designers. "*[It is] the process of narrowing down intents to perform precisely fit-for-purpose tasks. [It is] the same function but transforming what can be shown on the app screen into how the flows need to be organized. This is the differentiation point of [flow] scenario design from app design*" (D6).

Due to the constraints of the existing backend infrastructure and the characteristics of CUX, which follows a significantly flexible flow-based design with a relatively high degree of freedom, designers must execute repetitive operations to determine the optimal conversational UX. Verifying and eliminating duplicate intentions, which are stated as 'tangled,' are examples of repetitive tasks during the inspection of several entities that should be included in a single intent without omission. In the words of some of those interviewed, "*Determining 'which part is tangled' is a difficult task. Yes, in particular, it would be excellent if it could be identified. I am required to open all the logs to identify it*" (D11). "*There are numerous human errors in arranging the branches for each intent in a unified manner. I think there should be an automated tool*" (D4).

In the process of determining the ideal flow-based UX while maintaining the existing form-type service, the latter becomes a constraint, or sometimes, it leads to an adjustment to the form-type service that occurs during the development process. "*It was about transforming all of the app's functionality into a chatbot. Original chatbots were offered as a rule base. [...] So when we reorganized, we blasted everything apart and rebuilt it. We updated both the app and the website during the development process. As a result, (the development process) took a very long time*" (D14).

4.2 Lack of Common Design Patterns and Understanding of Feasibility

4.2.1 Lacking Standardized, What You See Is What You Get (WYSIWYG)-style Design Patterns. In the case of web-based and mobile applications employing GUI-based components, the most effective forms of components were established in the form of a standard or pattern over a long period of use, and the pattern is visual and intuitive in WYSIWYG format. During the design process, UX designers can easily predict how the design outcome will behave. In addition, developers have a common understanding of the results of implementation using the design outcome.

CUX design employs design components known as intent and entity, which are visually difficult to assess, and design standards are yet to be established. Designers must use these intents and entities to convert a form-based design with UI elements, such as radio buttons and checkboxes, into a flow-based design with natural language. Participants also agreed on the point about platform restrictions making it difficult to immediately examine the implementation, which leads to repetitive work.

4.2.2 Diverse Perspectives on Best CUX Design Practices. The CUX design process considers the form-to-flow transition and flow-appropriate UX. In this stage, the optimal UX is generated by combining intents and entities that are difficult to design in the intuitive WYSIWYG style. Participants typically identified the appropriate solutions via trial and error, and this was considered a stage wherein there was no general standardization. Tool software companies such as Google, Apple, Amazon, and Naver, which supply DMS, provide rough guidelines. However, a detailed standard as observed at the mobile application or web level [16] is lacking. "*In the case of the mobile application, and web-based platform, there is a consensus in that significant advancements have been achieved over time; however, perspectives on chatbots differ*" (D9). "*I suppose this represents the transitory period before a common mental model is developed. [...] In the case of chatbots, it is evident that a standard* *mental model is yet to be developed*" (D6). "*It appears challenging as the user experience can be altered based on how the intent is fragmented*" (D3).

4.2.3 Uncertainty in Development Feasibility. Compared with the mobile application/web platforms, CUX design involves different design components, and it is difficult to intuitively verify their operations. Furthermore, most CUX designers assume that CUX work is a process of bridging the gap between design and implementation, which development could not achieve. Some designers expressed this as a detour (D11), or a method of handling an infinite number of cases (D7, D11), whereas other designers described it as a method of identifying an alternate UX within the scope of development (D1, D4, D14). "*Regarding mobile applications and web, there is a certain degree of consensus between developers and designers that it is feasible to develop*" (D5). "*The problem is that, in the absence of the designer, it cannot be determined if the flow operates as intended. Only the designer can achieve this. [...] Despite the fact that I collaborated with engineers, they questioned me regarding several aspects*" (D14).

One of the responsibilities of CUX designers is to persuade developers to implement their designs. However, the participants encountered various challenges during this process. For example, there was generally minimal accumulated UX data that could be used as a basis for persuasion, and the development timeline was stringent. Moreover, developers were unfamiliar with the various implementation methodologies because DMS engines are also evolving. Several designers even experienced that being forced them to make design changes because of DMS version updates. "*Chatbot expertise is sparse. Hence, we should take trial and error*" (D7). "*We need to devise lots of detour tactics to overcome the engine's limitations. This procedure demands considerable effort*" (D11).

A critical stage of the CUX design process is the accumulation of training sentences for an NLP engine. However, there is a limitation, in that it is difficult to directly predict outcomes. The testing procedure employing an NLP engine trained using real-world text samples is limited; hence, tests are generally conducted using the Wizard of Oz technique.

4.2.4 Repetitive Work. Compared with mobile application/web platforms, the CUX process is characterized by a greater number of iterative tasks, necessitating a return to the previous phase and rework, for reasons such as the definition of standards, securing of data, consideration of designs observed during testing, and constraints of legacy systems. Such repetitive labor occurs before and after the service launch, and poses challenges to the CUX design process, given the numerous maintenance tasks and the required continual maintenance of the system after the service launch. A designer expressed the difficulty of repetitive work as follows. "*Why is this an AI system? All procedures were performed manually. It is manual labor performed by humans for AI*" (D10).

The repetitive tasks occurred during error correction. In the AI/ML-based CUX development environment, it was difficult to support the common trial-and-error process of the designer; thus, owing to repetitive manual work, the operation differed from the design intent. Errors in the intent recognition stage that occur due to overlapping intents expressed as "tangled" or due to missing entities that should have been included in one intent are typical examples. "*Currently, there are no rules or guidelines for resolving tangled errors, simply attempts, which are re-implemented if they are unsuccessful*" (D11). "*Conflicts may arise if utterances for each service are processed separately. The utterance collision and intent collision should be re-evaluated and revised*" (D1).

This repetitive work occurred during the creation and maintenance of the template documents for collaboration. The causes of repetitive document work are as follows. When the same content is processed in numerous document formats in various manners, such that stakeholders from diverse backgrounds can view the same content, this is considered as redundant work. Redundant documentation makes it difficult to manage collaboration in a coherent manner. "*To fulfill the requirements for chatbots, engine developers, and front-end developers, all the interactions should be* *arranged for each screen unit*" (D10). "*Therefore, while collaborating with multiple people, if they are not unified at the center, there will be a combination of distinct versions*" (D2).

4.3 Complexity of Collaborative Work Environments

4.3.1 Complexity and Challenges Induced by Evolving Technology. The CUX designers who participated in the interview stated that they could carry out design only if they understood the characteristics of the builder and platforms; however, they generally used multiple platforms simultaneously, and these platforms were constantly changing. It was suggested that it would be cumbersome for designers to work with a platform that was complex or had not been previously handled. "*What we can provide is dependent on the performance of the builder. Builder characteristics and NLP quality levels are constantly changing, given the current industry context. It is a chaotic moment!*" (D4).

4.3.2 Lack of Common Understanding among Stakeholders. The use of natural language familiar to humans is a significant factor in the development of CUX services. This can create an illusion for the stakeholders. The prevalent misconception that current application or web platform services can be readily converted into CUX services and that they can be developed by adding sentence learning that fits the service domain on top of a common DMS platform can lead to an underestimation of the amount of work required. Due to a lack of awareness regarding platform characteristics, strict development deadlines, and an insufficient workforce may result in a delayed service launch. Except for backend development, mobile applications, and web development are almost standardized and are performed in an environment where it is easy to use previously developed modules. In a relatively new development environment, applications are developed without a complete understanding of the platform's characteristics. In addition, perceptions of technological limitations may differ. Feasibility should be considered in conjunction with time and workforce resources. The time required to become accustomed to a new environment and the effort required to handle unanticipated issues in an environment where the expertise and knowledge required for modification are not generalized can limit the scope of implementation. The effort required to be made by the CUX designer to persuade stakeholders to implement the CUX design in this process may result in tensions within the organization. "*Chatbots require a significant amount of work to set the standards,*" as conveyed by participant D7.

4.3.3 Tension and Persuasion in Collaborative Design and Development. With respect to communication among stakeholders, there are several methods in the implementation stage–including persuasion–that can be used to achieve an optimal tradeoff between requirements and available resources. In an environment where stable technology and standardized implementation are common, there is a tacit consensus regarding this tradeoff. However, the communication process involved in determining the optimal tradeoff by persuading groups at different levels in the hierarchy is one of the factors influencing decision making. Further, constraints related to an unfamiliar environment may result in tensions within the organization. It is difficult to explain why effort is required to identify a new solution when no applicable alternatives are available, particularly for a CUX designer with minimal knowledge of development. "*Asking developers to create something which seems impossible to implement poses many challenges; when finding appropriate reference points to convince them is difficult, persuasion is also made more difficult*" (D5).

The interviews indicated that the majority of service development environments develop services requested by clients based on specific platforms such as Google, Amazon, and Naver. Moreover, only CUX designers among the interviewees had prior experiences in joining teams for constructing specialized platforms (D1). Therefore, this was not the general case. Due to the nature of CUX, it is required to have a developer with knowledge of AI and a data engineer for data learning and structure management. However, most teams do not have people with such skills; therefore, UX designers and developers should collaborate to solve all the potential challenges that arise during CUX design and development. There are cases of large corporations with developers or data engineers with AI expertise; however, in most cases, UX designers and developers are primarily involved in the service design, development, and evaluation stages. Developers have different levels of expertise with respect to specific builders and an understanding of the AI engine; thus, those with the most expertise should play the required roles.

4.3.4 Design Materials for Collaborative CUX Design. During the design phase of a mobile application or website, specialized tools are commonly employed for the construction of high- and low-fidelity prototypes, such as Figma or Sketch. Using these prototype tools for development via the process of transferring design results is common [53, 62, 73]. The generation and management of design documents may be burdensome for CUX designers in an environment where CUX design methodologies are not standardized. Most designers use Microsoft Excel-based documents to explain the conversation flow composed of intents and entities, in addition to current screen-based prototype-type documents, using a variety of documentation tools such as slide-based Microsoft PowerPoint presentations or Figma.

Given the flow-based characteristics of CUX, such redundancy in work is interpreted as a process in which the industry requires a documentation method suitable for expressing the desired information, such as generating different versions of blueprints, perspectives, and aerial views based on the intended architectural output. In the case of Microsoft Excel-based documents, the information type selected at the start of planning frequently changes during the development phase, thus necessitating multiple rounds of reworking. Additionally, in an environment where multiple builders are employed simultaneously, a Microsoft Excel document should be input for each builder. The CUX designer is assigned the additional task of re-adjusting the documents based on the characteristics of the builder.

4.4 Lack of Clarity on the Role of UX Designers

4.4.1 Absence of Consensus on Designers' Roles. A lack of commonality in understanding among the multiple stakeholders involved in the development of CUX services results in a lack of consensus on roles and responsibilities with respect to development. This results in lack of development of standard modules due to the absence of a common standard, issues with communication due to differences in the common understanding of feasibility, and numerous documents being generated for communication. Roles that did not previously exist, such as the structuring of training sentences, review of API interface connections, development of alternatives to overcome engine limitations, and securing domain-specific knowledge, were assigned.

New areas that did not previously exist (but were required) were considered as gray areas wherein the roles and responsibilities were unclear. CUX designers who frequently assumed the responsibilities of planning and program management were frequently allocated this role. "*The limits of the development environment and technology should be overcome by UX, which cannot cover all aspects*" (D6).

4.4.2 Perception of Gap Filling. In contrast to application/web development, where standards for planning and development have been established, a common consensus has yet to be established in CUX development. At a stage when technology is currently advancing, CUX designers are attempting to deliver the optimal UX in an environment that operates differently from their level of expectation. It was verified that this was an additional role to be played.

Among these additional roles, there were multiple processes required to bridge the gap between design and implementation, which development could not bridge, and it was confirmed that CUX designers play a role in bridging these gaps. Designers expressed detours in (D11) and identified methods for handling an infinite number of cases (D7). Moreover, designer D11 explained the role of UX in terms of identifying alternatives within the development scope to address the limitations of the engine.

Based on the interviews, the gap may be roughly categorized into two types: role recognition and development and implementation. Furthermore, the responsibility associated with delivering an optimal UX is frequently assigned to CUX designers. "*The generation of training data requires considerable effort and time. Ideally, this should be handled by a dedicated team. However, designers solely play this role in actual circumstances*" (D11).

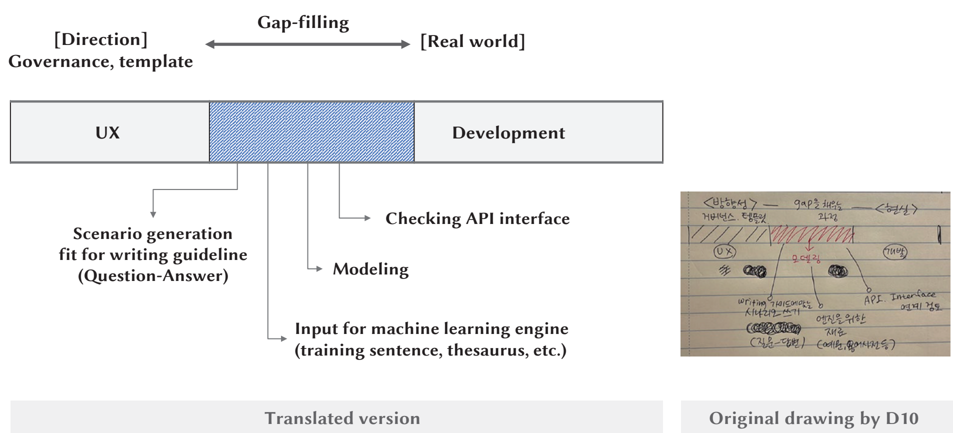

Fig. 6. Example of explanation for gap filling

D10 explained that the most important element of gap filling was modeling; the roles included in this modeling stage, namely the question-and-answer scenario writing, securing materials for the engine such as example sentences and terminology, and the API check for linking with the operating services, were explained. Further, it was explained that though the API check was not impossible to develop, there were certain situations in which it was difficult to create a separate API exclusively for CUX in an environment where services of various channels were provided using one API, that and this also acted as a new constraint in CUX design.

4.4.3 Expanded Role of UX Designers with AI/ML. We found that the performance of AI/ML in Natural Language Understanding (NLU), which plays a critical role in the CUX service, significantly altered designer perceptions. These changes influenced the responsibilities and roles of the designer. The quotes below illustrate scenarios in which the number of sentences used as examples for each intent should be balanced; otherwise, the intent training sentences would be reflected inaccurately. "*During actual development, I observed that if there are few example phrases in one location and numerous instances in another location, the outcome occasionally moves toward the location with the most examples, thus, influencing the value of the result. Therefore, the number of sample sentences per intent should be comparable; for example, 30 training sentences per intent. These types of rules are required*" (D7).

As an extension of the designer's role previously described, CUX designers are responsible for generating sample sentences, engine training, and verifying outcomes. In addition, several CUX designers were considering the necessity for additional expertise and roles to improve UX with respect to the characteristics of the dialogue. "*I am uncertain as to which expert should handle* *this; however, following discussions with engineers, a linguist or an AI specialist should be assigned independently*" (D14).

Even in a technical area where developers have sufficient expertise and responsibility, it may be challenging to set the standard for intent or adjust the settings to improve the quality of the system and user satisfaction based on whether it is acceptable from the user perspective instead of technical interpretation. Moreover, CUX designers are participating in these scenarios and suggesting alternative problem-solving strategies, as dedescribed earlier. "*For example, an outcome is considered as a false negative when it falls below a set threshold. If the chatbot is unable to detect it, the log is checked directly by a human. [...] It is critical to alter items that may clash with current intents and filter out these tasks while identifying false negatives*" (D4).

5 DISCUSSION

5.1 Challenges and Roles of CUX Designers in Multi-Channel Service Environments

5.1.1 Challenges in Form-to-Flow Transformation. We focused mainly on CUX design that considers the transition from 'form'-based interaction composed of visible objects known as a GUI to 'flow'-based interaction composed of intents, entities, and branches that lack visual affordance. This form-to-flow conversion frequently occurs in the process of providing multi-channel application services and is used in various applications such as travel reservations, banking transactions, and food delivery services [60]. Based on an analysis of in-depth interviews with CUX designers, it was confirmed that the CUX design process contained multiple knowledge-entry barriers.

In the case of form-based GUI interactions, there is a clear top-level navigational flow in each form which is designed based on information architecture. However, a flow-type UX with a high degree of freedom associated with the natural language-based manipulation presents unique challenges different from those of current form-based UX designs. Branch design in flow management refers to the process of constructing a user-accessible flow path with branching and reply patterns and returning to the original flow when the user deviates from the system-designed path. In the absence of a correct mental model for flow recognition and branching, the branching design process, without the assistance of familiar design principles, was identified as a significant challenge faced by CUX designers. There are widely used CUX design guidelines [7, 22, 46, 74]; however, established standards for CUX design are still lacking in this area [19], and supporting tools for CUX designers are inadequate [47, 48]. Furthermore, additional studies are required on design approaches and tools that can facilitate form-to-flow transformation and methods that can be used to readily express flow-based interactions, such as branching design.

5.1.2 Collaborative Work under Complex and Unstable AI/ML Environments. AI/ML-based environments are complex and undergo numerous changes. The characteristics of a service using technology that is evolving and the development environment based on the platforms of other companies serve as engines for CUX service development. Wenger suggested a community of practice framework based on the concept of legitimate peripheral participation by experienced members in the learning process by engaging with them [38]. Collaboration between various experts in a relatively new field (interdisciplinarity) could result in methods of execution, supplementation, and development of tools or artifacts, which are associated with various opinions regarding individual roles [40]. In this process, those with more expertise (generally UX experts) are responsible for transferring design knowledge and experiences to other less experienced members. For CUX design, there is a lack of design standards, and a relatively large amount of related experience is required to implement empirical standards. These aspects inevitably act as an additional burden on CUX designers, who are tasked with the design of an optimal UX for flows.

When designers develop mobile applications or websites, there is a well-established understanding of feasibility, role and responsibility, and human resource allocation. However, in CUX development, there is a lack of experienced designers and developers due to entry obstacles and a general lack of expertise, and this makes manpower allocation challenging. Furthermore, there are various perspectives on the practicability or feasibility of flow design and implementation. Moreover, in an unfamiliar context, it is difficult to explain why decision criteria are highly important to various stakeholders and to clearly provide an applicable alternative. Implementing new intent-level design scenarios for improved UX to meet user demands involves persuading developers, which leads to novel challenges for CUX designers.

Feasibility leads to decisions regarding the appropriate tradeoff between requirements and available resources. In the absence of standardized work, boundary-negotiating artifacts refer to objects required for collaboration between people with different roles in the collaborative exercise [39]. Intents, entities, and spreadsheets leading to their flow are yet to be established as industry-wide standards, although they are the most frequently used artifacts within development teams. AI introduces new challenges for designing interactions with AI-enabled services. There are several guidelines for human-AI interface design, mostly from large technology companies (e.g., Apple [31], Google [24], IBM [30], and Microsoft [5]). An ML-based DMS has a high degree of uncertainty between the input and output. Prior studies [32, 33, 68–71] focused on the issues encountered by UX designers in AI/ML environments when their designs did not perform as intended in real-world scenarios. According to a recent study by Yang et al. [69], AI technology is challenging for UX designers to understand and adopt within the context of evolving and adapting systems in the industry, in addition to AI/ML-based systems already being complex. In particular, this refers to novel UX and research-based techniques that are currently under development by engineers. Further, conversational agents are increasingly embedded into everyday persuasive services that involve AI/ML-based context sensing and adaptation [34, 41, 42]. Hence, it was suggested [69, 71, 72] that the existing design approaches, tools, and processes of designers within AI/ML environments should be augmented and improved.

Within the framework of collaboration in complex environments, CUX designers face two distinct challenges. The first is the lack of a standard design pattern. The second is the management of the collaborative process to discuss the feasibility of implementation in an environment that includes uncertainties resulting from the use of AI/ML, and reaching a consensus for the purpose of implementation of the desired UX.

5.1.3 Tech-Oriented Multidisciplinary Teams. It was revealed that the CUX designers who participated in the interviews volunteered or were compelled to assume various additional responsibilities during the CUX design process. The majority of these responsibilities fell into two categories: 1) the process of bridging the gap between design and implementation results that development could not provide, and 2) the process of bridging the gap in role responsibilities.

Designers use phrases such as "*finding tricks*," "*finding detours*," and "*filling*" to describe the process of bridging the gap between design and implementation. Moreover, they voluntarily collaborate with developers to enhance the UX. The designer's role is evolving from the traditional role of a producer-centered functional designer to that of a design facilitator and strategy leader [18]. According to a study by Kang et al. [35], the recent trend of developing new services based on innovation and technology in a multidisciplinary development environment emphasizes interdisciplinary collaboration in the early stages of design.

With respect to the gaps in the perception of roles, we identified the differences in the performers' impressions of additional roles, as derived from the AI/ML properties of NLU, which play a key role in the CUX service. Various additional roles, such as generating example (training) sentences for

ML or evaluating model outcomes, have not been previously reported. These additional tasks can be divided into those that require the assistance of specialists, such as data engineers, and those that require simple skills, such as crowd workers. These responsibilities are frequently handled by the CUX designers; however, in reality, such activities may require separate staff with expertise.

5.2 Needs for Added Support for CUX Designers

5.2.1 Supporting Form-to-Flow Design. For multi-channel environments, a design methodology for form-to-flow conversion that is optimized for the process of building a CUX service that integrates existing mobile applications and websites is required. Unlike form design, flow design involves a large design space due to the high degree of freedom of the operating method. As a design methodology, we propose standardized collaboration-enabling document templates as a design tool. As discussed earlier, we observed that different workers are required to collaborate on a large-scale CUX project; however, document reworking occurs due to differences in experience and perspective. The results revealed that during their process, skilled CUX designers repeatedly recollected all the material and redistributed the templates. Moreover, it was acknowledged that this occurs as the project advances due to the unstable development environment and processes. The design output is transformed into a document that is used as a medium to convey ideas to development professionals from various professional domains and roles, such as designers, developers, and customers, and to establish the scope of development. These design artifacts can be considered as boundary negotiating objects required for collaboration between individuals with varied functions in the absence of standardized work [39]. The documentation of well-defined design artifacts can facilitate team collaboration. We observed a significant amount of redundant documentation efforts directed toward the CUX designers, as reported in a related study [55]. A design tool that facilitates both designer expression and developer interpretation can facilitate collaboration between designers and developers in a complex environment [44]. In addition, frequently used CUX design strategies, such as error recovery strategies [10] and reply methods, are used as design patterns to aid the design process.

5.2.2 Supporting Functional and Empirical Testing for CUX Designers. Provisioning testing tools helps simplify the design and testing processes. These tools include 1) functional testing that helps to automatically identify errors, which occur due to duplication and omission, in advance, as is common in the design process of flow, and 2) empirical testing to determine whether the system is implemented as intended by the designers and whether the users can use the service properly when it is deployed.

In the context of functional testing, a typical error occurs in the intent recognition step, where it is difficult for the designer to recognize duplication or omission among a large number of intents. In this case, the application of automated applications is expected to be simple and efficient. Thus, CUX design, which starts with the definition of intents and entities, should enable CUX designers to readily understand changes based on the circumstances, such as the expansion of tasks in module units to the entire service, in addition to changes in usage or functionalities. From the most basic assistance that checks for overlapping intents in the sharing of current usage status via the dashboard, for example, the examination of the most frequently used phrases by users, interdisciplinary teams may accumulate data and make appropriate judgments during the collaboration process.

Empirical testing considers the possibility that chatbots may operate differently from the intent for which they were designed. However, in the CUX development environment based on ML, it is difficult to confirm the development results, and thus, designers rely on the trial-and-error approach. In numerous studies [69–72], limited prototyping was identified as one of the challenges in AI/ML-based UX design. Traditional HCI is an iterative strategy employing simple prototypes (e.g., spiral approach) that promote the understanding of complex systems or expected user reactions via Wizard of Oz prototyping. However, adaptive AI/ML technology based on big data does not guarantee the same results as laboratory-based small-scale prototyping when applied to a large-scale service building [70]. The system's mode of operation can be supplemented and enhanced via proactive management after its launch. Further, the technical implementation should be tested within a context with a large amount of data collected from the actual environment, and the design process should be modified such that UX practitioners can participate in this process.

5.2.3 Defining Extended Roles with Collaborative Perspectives. Unlike the traditional GUI-based UX approach, CUX designers are expected to play roles that were not previously considered. They are required to have fundamental knowledge of intent and entity concept, error repair concept, and CUX-optimized design principles. There are also unique technical characteristics and work environments related to NLP, such as generating training sentences, AL/ML-engine training, and tone and manner design in conversational interactions. It is difficult to provide a simple and complete solution to such complex and unfamiliar work contexts. Instead, our findings highlight that at the team-building stage for CUX design, it is important to recognize the work function additionally required in CUX design and the need to secure the professional competency required to fulfill the role In addition, it is expected that several of these new roles, such as an understanding of the necessity for integrated knowledge from numerous disciplines and problem-solving via collaboration with multiple stakeholders, will lead to a change in organizational culture.

6 CONCLUSION

Although numerous studies have been conducted on CUX design, studies on the real-world challenges faced by UX designers who work on multi-channel service design are limited. Based on the interviews with designers at actual industrial workplaces, this study revealed the overall process of form-to-flow-based CUX design and identified four critical challenges encountered by CUX designers: (1) differences in design processes and activities, (2) a lack of common design patterns and understanding of feasibility, (3) complexity of collaborative work environments, and (4) a lack of clarity on the role of CUX designers. We explained these challenges in the context of changed work environment changes, such as collaborative work with technology-oriented multidisciplinary teams under complex and unstable AI/ML environments. Based on these findings, we discussed the needs for added support for CUX designers, including tools and guidelines to support form-to-flow design, tools for testing, and defining extended roles of CUX designers with collaborative perspectives. We anticipate that this study would catalyze the realization of such collaborations and the development of newer supports.

ACKNOWLEDGMENTS

This research was supported by LGE-KAIST Digital Health Research Center (DHRC). The lead author of this work, J. Heo, is affiliated with KAIST and Kookmin University. We are grateful to Jaehee Park and Jieun Lee for supporting parts of the illustration work.

REFERENCES

[1] Ahmad Abdellatif, Diego Costa, Khaled Badran, Rabe Abdalkareem, and Emad Shihab. 2020. Challenges in Chatbot Development: A Study of Stack Overflow Posts. In *Proceedings of the 17th International Conference on Mining Software Repositories (MSR '20)*. ACM Press, New York, NY, USA, 174–185. https://doi.org/10.1145/3379597.3387472

[2] Eleni Adamopoulou and Lefteris Moussiades. 2020. An Overview of Chatbot Technology. In *Artificial Intelligence Applications and Innovations (AIAI 2020, IFIP Advances in Information and Communication Technology)*. Springer International Publishing, Cham, 373–383. https://doi.org/10.1007/978-3-030-49186-4_31

[3] Eleni Adamopoulou and Lefteris Moussiades. 2020. Chatbots: History, Technology, and Applications. *Machine Learning with Applications* 2 (2020), 100006. https://doi.org/10.1016/j.mlwa.2020.100006

[4] Christopher Alexander. 1977. *A Pattern Language: Towns, Buildings, Construction.* Oxford university press.

[5] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19).* ACM Press, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300233

[6] Deniz Arsan, Ali Zaidi, Aravind Sagar, and Ranjitha Kumar. 2021. App-Based Task Shortcuts for Virtual Assistants. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21).* ACM Press, New York, NY, USA, 1089–1099. https://doi.org/10.1145/3472749.3474808

[7] Google Assistant. 2021. *Interactive Canvas Design Guidelines.* Retrieved April 15, 2023 from https://developers.google.com/assistant/df-asdk/interactivecanvas/design/guidelines

[8] Anna Bagnoli. 2009. Beyond the Standard Interview: The Use of Graphic Elicitation and Arts-Based Methods. *Qualitative Research* 9, 5 (2009), 547–570. https://doi.org/10.1177/1468794109343625

[9] Keith C Barton. 2015. Elicitation Techniques: Getting People to Talk about Ideas They Don't Usually Talk about. *Theory & Research in Social Education* 43, 2 (2015), 179–205. https://doi.org/10.1080/00933104.2015.1034392

[10] Dennis Benner, Edona Elshan, Sofia Schöbel, and Andreas Janson. 2021. What Do you Mean? A Review on Recovery Strategies to Overcome Conversational Breakdowns of Conversational Agents. In *International Conference on Information Systems (ICIS).* 1–17.

[11] Jan O Borchers. 2000. A Pattern Approach to Interaction Design. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '00).* ACM Press, New York, NY, USA, 369–378. https://doi.org/10.1145/347642.347795

[12] Petter Bae Brandtzaeg and Asbjørn Følstad. 2018. Chatbots: Changing User Needs and Motivations. *Interactions* 25, 5 (aug 2018), 38–43. https://doi.org/10.1145/3236669

[13] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.

[14] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: Multi-"touch" Interaction around Small Devices. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08).* ACM Press, New York, NY, USA, 201–204. https://doi.org/10.1145/1449715.1449746

[15] David Cancel and Dave Gerhardt. 2019. *Conversational Marketing: How the World's Fastest Growing Companies Use Chatbots to Generate Leads 24/7/365 (and How You Can Too).* John Wiley & Sons.

[16] Ana Paula Chaves and Marco Aurelio Gerosa. 2021. How Should My Chatbot Interact? A Survey on Social Characteristics in Human-Chatbot Interaction Design. *International Journal of Human-Computer Interaction* 37, 8 (2021), 729–758. https://doi.org/10.1080/10447318.2020.1841438

[17] Nancy Chen, Mike Chen-ho Chao, Henry Xie, and Dean Tjosvold. 2017. Transforming Cross-cultural Conflict into Collaboration: The Integration of Western and Eastern Values. *Cross Cultural & Strategic Management* 25, 1 (2017), 70–95. https://doi.org/10.1108/CCSM-10-2016-0187

[18] Seung Jin Chung, On Shim, and Soyeon Kim. 2022. Analysing the Status of Communication Designers' Role and Interdisciplinary Collaboration. *The Design Journal* 25, 3 (2022), 397–415. https://doi.org/10.1080/14606925.2022.2058682

[19] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, Vincent Wade, and Benjamin R. Cowan. 2019. What Makes a Good Conversation? Challenges in Designing Truly Conversational Agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19).* ACM Press, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300705

[20] Nathan Crilly, Alan F Blackwell, and P John Clarkson. 2006. Graphic Elicitation: Using Research Diagrams as Interview Stimuli. *Qualitative Research* 6, 3 (2006), 341–366. https://doi.org/10.1177/1468794106065007

[21] Ticianne Darin, Nayana Carneiro, Rossana Andrade, Rute Castro, and Rodrigo Almeida. 2019. Integrating HCI Perspective into a Mobile Software Development Team: Strategies and Lessons from the Field. In *Proceedings of the 21st international conference on human-computer interaction with mobile devices and services (MobileHCI '19).* ACM Press, New York, NY, USA, 1–10. https://doi.org/10.1145/3338286.3347545

[22] Amazon Developer. 2017. Designing for Voice - Design Checklist. Retrieved April 15, 2023 from https://developer.amazon.com/fr/designing-for-voice/design-checklist

[23] David Gauntlett. 2007. *Creative Explorations: New Approaches to Identities and Audiences.* Routledge. https://books.google.co.kr/books?id=yyp9AgAAQBAJ

[24] Google. 2020. *The People + AI Research (PAIR) Guidebook.* Retrieved April 15, 2023 from https://pair.withgoogle.com/guidebook/

[25] David Griol, Javier Carbó, and José M Molina. 2013. An Automatic Dialog Simulation Technique to Develop and Evaluate Interactive Conversational Agents. *Applied Artificial Intelligence* 27, 9 (2013), 759–780. https://doi.org/10.1080/08839514.2013.835230
[26] Greg Guest, Kathleen M MacQueen, and Emily E Namey. 2011. *Applied Thematic Analysis.* sage publications.
[27] Chris Harrison, Julia Schwarz, and Scott E Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. ACM Press, New York, NY, USA, 627–636. https://doi.org/10.1145/2047196.2047279
[28] Hendrik Heuer and Daniel Buschek. 2021. Methods for the Design and Evaluation of HCI+ NLP Systems. arXiv:2102.13461
[29] Matthew K. Hong, Adam Fourney, Derek DeBellis, and Saleema Amershi. 2021. Planning for Natural Language Failures with the AI Playbook. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM Press, New York, NY, USA, Article 386, 11 pages. https://doi.org/10.1145/3411764.3445735
[30] IBM. 2022. IBM Designing for AI. Retrieved April 15, 2023 from https://www.ibm.com/design/ai
[31] Apple Inc. 2021. *Introduction to Machine Learning - Apple Human Interface Guidelines.* Retrieved April 15, 2023 from https://developer.apple.com/design/human-interface-guidelines/technologies/machine-learning/introduction
[32] Jinglu Jiang, Surinder Kahai, and Ming Yang. 2022. Who Needs Explanation and When? Juggling Explainable AI and User Epistemic Uncertainty. *International Journal of Human-Computer Studies* 165 (2022), 102839. https://doi.org/10.1016/j.ijhcs.2022.102839
[33] Ridong Jiang and Rafael E. Banchs. 2017. Towards Improving the Performance of Chat Oriented Dialogue System. In *2017 International Conference on Asian Language Processing (IALP)*. IEEE, 23–26. https://doi.org/10.1109/IALP.2017.8300537
[34] Yugyeong Jung, Gyuwon Jung, Sooyeon Jeong, Chaewon Kim, Woontack Woo, Hwajung Hong, and Uichin Lee. 2023. "Enjoy, but Moderately!": Designing a Social Companion Robot for Social Engagement and Behavior Moderation in Solitary Drinking Context. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023). https://doi.org/10.1145/3610028
[35] Hyo-Jin Kang, Kyung-won Chung, and Ki Young Nam. 2015. A competence Model for Design Managers: A Case Study of Middle Managers in Korea. *International Journal of Design* 9, 2 (2015).
[36] Lorenz Cuno Klopfenstein, Saverio Delpriori, Silvia Malatini, and Alessandro Bogliolo. 2017. The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*. ACM Press, New York, NY, USA, 555–565. https://doi.org/10.1145/3064663.3064672
[37] Christian Kruschitz and Martin Hitz. 2010. Are Human-Computer Interaction Design Patterns Really Used?. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. ACM Press, New York, NY, USA, 711–714. https://doi.org/10.1145/1868914.1869011
[38] Jean Lave and Etienne Wenger. 1991. *Situated Learning: Legitimate Peripheral Participation.* Cambridge university press. https://books.google.co.kr/books?id=CAVIOrW3vYAC
[39] Charlotte P Lee. 2007. Boundary Negotiating Artifacts: Unbinding the Routine of Boundary Objects and Embracing Chaos in Collaborative Work. *Computer Supported Cooperative Work (CSCW)* 16, 3 (2007), 307–339. https://doi.org/10.1007/s10606-007-9044-5
[40] Charlotte P. Lee and Drew Paine. 2015. From The Matrix to a Model of Coordinated Action (MoCA): A Conceptual Framework of and for CSCW. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM Press, New York, NY, USA, 179–194. https://doi.org/10.1145/2675133.2675161
[41] Uichin Lee, Kyungsik Han, Hyunsung Cho, Kyong-Mee Chung, Hwajung Hong, Sung-Ju Lee andYoungtae Noh, Sooyoung Park, and John M. Carroll. 2019. Intelligent Positive Computing with Mobile, Wearable, and IoT Devices: Literature Review and Research Directions. *Ad Hoc Networks* 83 (2019), 8–24. https://doi.org/10.1016/j.adhoc.2018.08.021
[42] Uichin Lee, Gyuwon Jung, Eunyeol Ma, Jinsan Kim, Heepyung Kim, Jumabek Alikhanov, Youngtae Noh, and Heeyoung Kim. 2023. Toward Data-Driven Digital Therapeutics Analytics: Literature Review and Research Directions. *IEEE/CAA Journal of Automatica Sinica* 10, 1 (2023), 42–66. https://doi.org/10.1109/JAS.2023.123015
[43] Toby Jia-Jun Li and Oriana Riva. 2018. Kite: Building Conversational Bots from Mobile Apps. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18)*. ACM Press, New York, NY, USA, 96–109. https://doi.org/10.1145/3210240.3210339
[44] Nolwenn Maudet, Germán Leiva, Michel Beaudouin-Lafon, and Wendy Mackay. 2017. Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM Press, New York, NY, USA, 630–641. https://doi.org/10.1145/2998181.2998190
[45] Michael McTear. 2018. Conversation Modelling for Chatbots: Current Approaches and Future Directions. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2018* (2018), 175–185.
[46] Robert J Moore and Raphael Arar. 2019. *Conversational UX Design: A practitioner's Guide to the Natural Conversation Framework.* Association for Computing Machinery and Morgan & Claypool Publishers.

[47] Robert J Moore, Eric Young Liu, Saurabh Mishra, and Guang-Jie Ren. 2020. Design systems for conversational UX. In *Proceedings of the 2nd Conference on Conversational User Interfaces (CUI '20)*. ACM Press, New York, NY, USA. https://doi.org/10.1145/3405755.3406150

[48] Cosmin Munteanu, Leigh Clark, Benjamin Cowan, Stephan Schlögl, María Inés Torres, Justin Edwards, Christine Murad, Matthew Aylett, Martin Porcheron, Heloisa Candello, Philip Doyle, and Jaisie Sin. 2020. CUI: Conversational User Interfaces: A Workshop on New Theoretical and Methodological Perspectives for Researching Speech-Based Conversational Interactions. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion (IUI '20)*. ACM Press, New York, NY, USA, 15–16. https://doi.org/10.1145/3379336.3379358

[49] Christine Murad and Cosmin Munteanu. 2019. Teaching for Voice: the State of VUI Design in HCI Education. In *Proceedings of EduCHI 2019 Symposium*.

[50] Christine Murad, Cosmin Munteanu, Benjamin R. Cowan, and Leigh Clark. 2019. Revolution or Evolution? Speech Interaction and HCI Design Guidelines. *IEEE Pervasive Computing* 18, 2 (2019), 33–45. https://doi.org/10.1109/MPRV.2019.2906991

[51] Christine Murad, Cosmin Munteanu, Benjamin R. Cowan, and Leigh Clark. 2021. Finding a New Voice: Transitioning Designers from GUI to VUI Design. In *Proceedings of the 3rd Conference on Conversational User Interfaces (CUI '21)*. ACM Press, New York, NY, USA, Article 22, 12 pages. https://doi.org/10.1145/3469595.3469617

[52] Harold G. Nelson and Erik Stolterman. 2012. *The Ultimate Particular*. MIT Press, Chapter 1, 27–40. https://books.google.co.kr/books?id=lr34DwAAQBAJ

[53] Erik G Nilsson. 2009. Design Patterns for User Interface for Mobile Applications. *Advances in Engineering Software* 40, 12 (2009), 1318–1328. https://doi.org/10.1016/j.advengsoft.2009.01.017

[54] Ketakee Nimavat and Tushar Champaneria. 2017. Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. *International Journal of Science and Research Development* 5, 7 (2017), 1019–1024.

[55] Fatih Kursat Ozenc, Miso Kim, John Zimmerman, Stephen Oney, and Brad Myers. 2010. How to Support Designers in Getting Hold of the Immaterial Material of Software. In *Proceedings of the 2010 CHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM Press, New York, NY, USA, 2513–2522. https://doi.org/10.1145/1753326.1753707

[56] David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–25. https://doi.org/10.1145/3449205

[57] Martin Porcheron, Joel E. Fischer, Stuart Reeves, and Sarah Sharples. 2018. Voice Interfaces in Everyday Life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM Press, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174214

[58] Sara Pérez-Soler, Sandra Juárez-Puerta, Esther Guerra, and Juan de Lara. 2021. Choosing a Chatbot Development Tool. *IEEE Software* 38, 4 (2021), 94–103. https://doi.org/10.1109/MS.2020.3030198

[59] A M Rahman, Abdullah Al Mamun, and Alma Islam. 2017. Programming Challenges of Chatbot: Current and Future Prospective. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. 75–78. https://doi.org/10.1109/R10-HTC.2017.8288910

[60] Jupiter Research. 2022. How Chatbots Will Facilitate the Omnichannel Experience. Retrieved April 15, 2023 from https://www.juniperresearch.com/document-library/white-papers/how-chatbots-will-facilitate-the-omnichannel

[61] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects. In *Proceedings of the 2012 CHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM Press, New York, NY, USA, 483–492. https://doi.org/10.1145/2207676.2207743

[62] Ahmed Seffah. 2010. The Evolution of Design Patterns in HCI: from Pattern Languages to Pattern-oriented Design. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems (PEICS '10)*. ACM Press, New York, NY, USA, 4–9. https://doi.org/10.1145/1824749.1824751

[63] Abu Shawar and ES Atwell. 2007. Chatbots: Are They Really Useful? *Journal for Language Technology and Computational Linguistics* 22, 1 (2007), 29–49.

[64] Hariharan Subramonyam, Jane Im, Colleen Seifert, and Eytan Adar. 2022. Solving Separation-of-Concerns Problems in Collaborative Design of Human-AI Systems through Leaky Abstractions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. ACM Press, New York, NY, USA, 1–21. https://doi.org/10.1145/3491102.3517537

[65] Yuan Tian, Ke Zhou, Mounia Lalmas, and Dan Pelleg. 2020. Identifying Tasks from Mobile App Usage Patterns. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR '20)*. ACM Press, New York, NY, USA, 2357–2366. https://doi.org/10.1145/3397271.3401441

[66] Zhiyuan Wan, Xin Xia, David Lo, and Gail C. Murphy. 2021. How Does Machine Learning Change Software Development Practices? *IEEE Transactions on Software Engineering* 47, 9 (2021), 1857–1871. https://doi.org/10.1109/TSE.2019.2937083

[67] Hironori Washizaki, Hiromu Uchida, Foutse Khomh, and Yann-Gaël Guéhéneuc. 2019. Studying Software Engineering Patterns for Designing Machine Learning Systems. In *2019 10th International Workshop on Empirical Software Engineering*

in Practice (IWESEP). IEEE, 49–495. https://doi.org/10.1109/IWESEP49350.2019.00017

[68] Marcus Winter and Phil Jackson. 2020. Flatpack ML: How to Support Designers in Creating a New Generation of Customizable Machine Learning Applications. In *Proceedings of the 2020 International Conference on Human-Computer Interaction, Design, User Experience, and Usability. (Lecture Notes in Computer Science, Vol. 700)*. Springer, Cham, 175–193. https://doi.org/10.1007/978-3-030-49760-6_12

[69] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating How Experienced UX Designers Effectively Work with Machine Learning. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18)*. ACM Press, New York, NY, USA, 585–596. https://doi.org/10.1145/3196709.3196730

[70] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. 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 (CHI '20)*. ACM Press, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376301

[71] Nur Yildirim, Alex Kass, Teresa Tung, Connor Upton, Donnacha Costello, Robert Giusti, Sinem Lacin, Sara Lovic, James M O'Neill, Rudi O'Reilly Meehan, Eoin Ó Loideáin, Azzurra Pini, Medb Corcoran, Jeremiah Hayes, Diarmuid J Cahalane, Gaurav Shivhare, Luigi Castoro, Giovanni Caruso, Changhoon Oh, James McCann, Jodi Forlizzi, and John Zimmerman. 2022. How Experienced Designers of Enterprise Applications Engage AI as a Design Materia. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. ACM Press, New York, NY, USA, Article 483, 13 pages. https://doi.org/10.1145/3491102.3517491

[72] Sabah Zdanowska and Alex S Taylor. 2022. A Study of UX Practitioners Roles in Designing Real-World, Enterprise ML Systems. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. ACM Press, New York, NY, USA, Article 531, 15 pages. https://doi.org/10.1145/3491102.3517607

[73] Cheng Zhang and David Budgen. 2011. What Do We Know about the Effectiveness of Software Design Patterns? *IEEE Transactions on Software Engineering* 38, 5 (2011), 1213–1231. https://doi.org/10.1109/TSE.2011.79

[74] Qingxiao Zheng, Yiliu Tang, Yiren Liu, Weizi Liu, and Yun Huang. 2022. UX Research on Conversational Human-AI Interaction: A Literature Review of the ACM Digital Library. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. ACM Press, New York, NY, USA, 1–24. https://doi.org/10.1145/3491102.3501855

Received January 2023; revised April 2023; accepted May 2023