If [YourName] Can Code, So Can You! End-User Robot Programming For Non-Experts

Manasi Swaminathan
Indiana University
Bloomington, IN, USA
mswamina@iu.edu

Long-Jing Hsu
Indiana University
Bloomington, IN, USA
hsulon@iu.edu

Min Min Thant
Indiana University
Bloomington, IN, USA
minthant@iu.edu

Kyrie Jig Amon
Indiana University
Bloomington, IN, USA
kjig@iu.edu

Anna S. Kim
Indiana University
Bloomington, IN, USA
kimas@iu.edu

Katherine M. Tsui
Toyota Research Institute
Cambridge, Massachusetts, USA
kate.tsui@tri.global

Selma Šabanović
Indiana University
Bloomington, IN, USA
selmas@indiana.edu

David J. Crandall
Indiana University
Bloomington, IN, USA
djcran@indiana.edu

Weslie Khoo
Indiana University
Bloomington, IN, USA
weskhoo@iu.edu

A	B	C	D	E
	Blocks	**Description**	**Gesture**	**Face**
	hello	introduce yourself using the bio	Default	Happy
	day	ask the user how their day is	Default	Random
	name	ask the user what their name is	Wave	Happy
	plans	ask the user what they'll do today	Wave	Excited
	goodbye	say goodbye to the user, wishing them good luck with their plans	Wave	Happy

Figure 1: Screen capture of the *Blocks* tab from the Google Sheets Interface of the Engine

ABSTRACT

Social robots are being studied for a wide variety of user populations, such as older adults, but programming these social robots typically requires deep technical knowledge. In this study, we developed a no-code end-user robot programming interface, with the goal of our interface being to empower individuals with no programming background to easily create social robot interactions with older adults using natural language. We evaluated five individuals with connections to adults older than 65 without robot programming experience. They were tasked with designing a simple conversation with the robot. We recorded their experiences using a survey and found that participants successfully used the interface to make the robot communicate with older adults. Overall, the participants found the interface easy to use and enjoyed the process. Thus, we provide recommendations on how to improve no-code end-user robot programming interfaces further.

CCS CONCEPTS

• **Applied computing** → **Sociology**; • **Computing methodologies** → **Natural language processing**; **Computer vision**; • **Human-centered computing** → **Interaction design**.

KEYWORDS

human-centered interaction design, conversational systems, robotics, machine learning, natural language processing

ACM Reference Format:
Manasi Swaminathan, Long-Jing Hsu, Min Min Thant, Kyrie Jig Amon, Anna S. Kim, Katherine M. Tsui, Selma Šabanović, David J. Crandall, and Weslie Khoo. 2024. If [YourName] Can Code, So Can You! End-User Robot Programming For Non-Experts. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion), March 11–14, 2024, Boulder, CO, USA*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3610978.3640644

HRI '24 Companion, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0323-2/24/03
https://doi.org/10.1145/3610978.3640644

1 INTRODUCTION

Social robots are gaining popularity and are now used in various settings such as schools and hospitals. Human-robot interaction

(HRI) research has increasingly focused on the older adult population, particularly exploring conversational robots designed to improve well-being [11, 19]. With numerous activities that social robots could engage in with older adults [13, 15], our goal was to develop a no-code end-user robot programming interface named ENGINE, for users such as caregivers who would like to program social robots for this demographic.

Programming a robot's behavior typically requires deep technical expertise — usually a background in computer science or engineering. This makes it almost impossible for the typical end-user to customize their robot's behavior. Our vision is to empower individuals with no programming background to program social robots easily for interactions with older adults. This could significantly improve the usability of social robots for this demographic, allowing, for example, a caregiver to customize a robot's behavior based on an older adult's specific preferences.

Toward this goal, we developed a system to allow programming a social robot's behavior using no code. The behavior is specified through natural language on an online spreadsheet (Google Sheets), allowing end-users to write the robot "programs" using a familiar interface. An interpreter running on the Robot Operating System (ROS) robot executes the instructions from the Google Sheets so that the user can make changes in real-time and see them immediately reflected on the robot. We leverage the recent rise of Large Language Models (LLMs), particularly the GPT family [23], to allow users to specify complex functionality in natural language.

2 RELATED WORK

2.1 Social Robots For Conversation

Many AI applications prominently use natural language-based interfaces, including mobile phones, websites, and virtual assistants such as Amazon Alexa and Microsoft Cortana [25]. More directly related to our context of social robots for older adults, Telenoid was designed to promote conversation with older adults living with dementia [22]. Other work has used social robots to conduct short autonomous conversations with older adults to help them reflect on what brings feelings of joy and meaning in life [12].

A major limitation in studies involving social robots is their inability to keep up with conversational topics [22]. Another study, which explored the benefits of social robots for older adults and dementia patients, suggested the need for more robust models capable of facilitating two-way conversations and handling the extreme emotions of this demographic [14].

2.2 No-Code End-User Robot Programming

Methodologies such as visual programming, augmented and mixed reality, demonstrations using kinetic teaching, speech-based programming, and tangible programming have been explored in industrial robotics [1]. However, there is still a need to make social robot programming accessible, given its growing popularity.

Visual programming has been successfully developed for social robots. One such study used Nao robots to interact with users based on gestures, where the reactions to specific gestures were specified using if-then-else blocks [8]. This methodology is still complex for non-programmers to understand as it uses the concept of conditional statements. To explore the use of natural language to program robots, a web-based tool called English2NAO was introduced, where users use simple English and include a dual representation (textual and formal) of the programming scenario [6]. However, this study was done without a robot, and users struggled to understand the program flow.

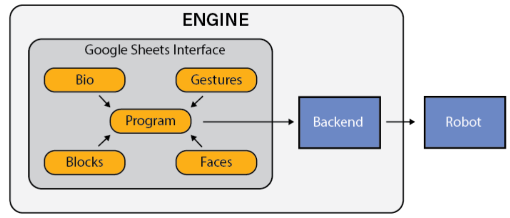

Figure 2: Flow diagram to show the relationship between components in the Engine. Arrows indicate one-way interaction between the components in the direction of the arrow.

2.3 Large Language Models

Large Language Models such as OpenAI's GPT-3 Application Programming Interface (API) and GPT-4 API, along with the ChatGPT web interface, have quickly revolutionized Natural Language Understanding and Generation across innumerable application areas [23]. In HRI, researchers have started to leverage these models' capabilities to mimic human-like conversational experiences with robots [24] such as NAO, Pepper, and Furhat [3, 7]. GPT-3.5's integration into Pepper showed promising applications in assisting individuals with Autism Spectrum Disorder[2]. With Furhat, GPT-3.5 helped generate smooth dialogue and emoticons. The emoticons were created based on sentiment analysis and then translating them into robot expressions [7].

Researchers have thoroughly evaluated prompt-based dialogue generation in large language models of different sizes and across many applications — education, business, creative writing, and programming [20]. The findings indicate that prompt-based generation effectively creates dialogue systems, especially in larger capacity language models [17]. LLMs' ability to convert natural language into code is predicted to revolutionize programming environments [9], including lowering the entry barrier for individuals with minimal programming expertise by, for example, enabling low-code tools such as visual programming.

3 METHODS & USER STUDY

3.1 Robot and The Engine

3.1.1 The Robot. Our framework is general enough to work on many social robots, but for this study, we use LuxAI's QTrobot (Luxembourg) [16], a tabletop humanoid robot that can move its arms and head (Fig 3A). It has a display for facial expressions, an internal microphone array, and speakers for recording and playing audio. At the time of the engine's development, versions of the robot lacked real-time chat communication features, such as GPT.

3.1.2 The Engine: The programming platform (which we call the "Engine") consists of two parts: a Google Sheets Interface in which users write their programs and a backend that executes them on the robot (Fig 2).

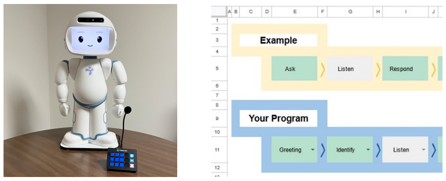

Figure 3: (A) Robot setup, including the external mic and control panel. (B) Screen capture of the *Program* tab.

Google Sheets Interface We used Google Sheets as a convenient and familiar interface for end-users to enter their "programs." Our backend running on the robot sees changes to the Google Sheets in real time; it consists of 5 specific tabs:

Bio in which the user writes a free-form, natural-language "biography" of the robot: its name, origin story, persona, etc. The participant edits this tab based on their creativity to give the robot its own story.

Program in which the conversation is designed using a flow chart-like visualization. The conversation is made of multiple blocks, and each block is a basic functionality that the robot does (say something, listen, etc.). The blocks are created in the *Blocks* tab and then selected in the flow chart to specify the conversation flow (Fig 3 B).

Blocks specify natural language instructions to the robot, and corresponding facial expressions and gestures. An instruction could be something like, "Ask the user a quiz question about animals, addressing the user by their name," or "If the user correctly answered the quiz question, then congratulate them; otherwise, give a supportive reply with the correct answer." This tab contains some predefined basic blocks that the users could use, shown in Fig 1.

Faces contains a list of facial expressions that the robot can do. Users can refer to this sheet to see the list of pre-programmed expressions. The robot allows users to create custom facial expressions, but in this study, we limited users to the pre-programmed ones.

Gestures is similar to the *Faces* tab, but contains the list of gestures (arm and head movements) that the robot can do.

The Backend: We used OpenAI's Whisper API [18] for transcribing audio, and OpenAI's GPT 3.5 text-davinci-003 model for generating dialogue through prompts. Prompts are created automatically by pulling information from the robot biography, the conversation history, and user-written instructions for the currently executing block. The user's instructions are written in a natural language format similar to communicating with ChatGPT.

This process leverages the "knowledge" of the language model (trained by OpenAI on a large corpus of text) to generate contextually relevant text based on text input [5]. For speaking, we use the robot's built-in text-to-speech capability. While the robot has a built-in microphone array, it is located on the top of the robot and thus is sensitive to environmental noise; we used an external USB microphone instead. We also added an external control panel (using an Elgato Stream Deck MK.2 (Munich, Germany) to allow the user to change the robot's volume and exit the system. Instead of solving the (difficult) problem of detecting when the user is speaking and when they are done speaking, users pressed and held a push-to-talk button on the Stream Deck.

3.2 User Study Procedures

In this study, we assessed the usability of the Engine's Google Sheet Interface. We evaluated the participants' ability to use and comprehend our no-code programming tool for enabling the robot to engage in simple conversations; we invited them to meet the robot in person, providing them with hands-on experience.

We first presented the Engine through a step-by-step instruction manual, followed by a 20-minute demonstration. We then asked the participants to create an activity for an imaginary older adult that included the robot introducing itself and asking at least two questions.

After the participants programmed the engine, researchers helped them run the program on the robot and refine their program until they were satisfied.

We also asked them to complete a Qualtrics-based survey to record their impressions on the task usability and load of the interface on a Likert scale of 1 to 5, with 1 being strongly disagree and 5 being strongly agree. To assess task usability, we employed the System Usability Scale [4, 21], with the following items (* indicates reverse-scoring): 1. *I think that I would like to do this task frequently.* 2. *I found the programming task unnecessarily complex* (*). 3. *I thought the programming task was easy to perform.* 4. *I think that I would need the support of a technical person to be able to perform the programming task* (*). 5. *I would imagine that most people would learn to do this programming task very quickly.* 6. *I felt very confident doing the programming task.* 7. *I needed to learn a lot of things before I could get going with the programming task* (*). 8. *I found the programming task very challenging to perform* (*).

To assess task load, we employed a modified version of the NASA Task Load Index [2, 10, 21], with the following items: 1. *How mentally demanding was the task*? 2. *How hurried or rushed was the pace of the task*? 3. *How successful were you in accomplishing what you were asked to do*? (*) 4. *How hard did you have to work to accomplish your level of performance*? 5. *How insecure, discouraged, irritated, stressed and annoyed were you during the task*?

The survey also collected demographic information about the participant's experience with coding and programming and their age. We interviewed the participants to better understand their experience with the tool and explored their preferences and dislikes.

Throughout the experiment, a researcher was present in the room, and the experiment was also recorded via Zoom and a voice recording device for a more thorough analysis of the participants' interaction and experience using the Engine. The experiment took 1 hour, and participants were each compensated with a US$30 gift card. We analyzed our data quantitatively and qualitatively. The quantitative task usability and task load results were statistically analyzed using Minitab (State College, USA). The qualitative interview results were transcribed and analyzed as we grouped the findings.

3.3 Participants

As the robot was designed for interaction with older adults, we sought individuals with connections to adults older than 65 years.

We recruited 5 participants between the ages of 21 and 43 through word of mouth (3 female, 2 male). Two participants had attended coding classes, but none had experience programming a robot.

4 FINDINGS

4.1 Quantitative Findings

First, we assessed the internal consistency of questions for task usability and task load separately, and both exhibited a high level of internal consistency (Cronbach's alpha= 0.884 and 0.807, respectively). Subsequently, our task usability score above the midpoint (mean = 3.65, SD = 0.55) suggests that our interface is usable, and our task load below the midpoint (mean = 2.48, SD = 0.36) suggests that our interface has a low mental load.

4.2 Qualitative Findings

Overall, all participants felt that the interface was easy to use. Two participants mentioned that the experience was fun. They liked the *"visual element of it(P2 (21/M),"* and appreciated the immediate feedback of *"being able to put in your blocks and get immediate feedback right away without taking too long* (P4 (29/M)," consistent with the high task usability score and low task score. Although some participants encountered challenges during their initial attempts, such as errors in prompt writing or block selection, most participants were satisfied with second iteration programs.

4.3 Navigating The Google Sheets Interface

Participants did not seem to experience difficulty toggling between the *Bio*, *Program*, and *Blocks* tabs of Google Sheets. Most participants did not spend much time on *Gestures* and *Faces*. P4 (29/M) said, *"No time to experiment with gestures and faces since the focus was trying to get the [instructions] right for the task."* P2 (21/M) felt that the currently available gestures did not make sense: *"you wouldn't be asking a question and then like, start crying or something or look disgusted?"* Similarly, P5 (29/F) forgot to insert 'Listen' blocks into her program and wondered if there is a way to shift blocks by one instead of having to redo the selection of the blocks.

4.4 Interaction With The Robot

Some participants did not fully understand the capabilities of the framework. For example, P1 (43/F) did not know about the repeat function and, therefore, did not incorporate it into the program. Consequently, when she requested the robot to *"Can you say that again?"* during testing, the robot failed to repeat and proceeded with its pre-programmed script. Most participants did not ask the robot questions when conversing with the robot because they thought the robot was limited by *"not being able to ask questions back."* They did not realize that the robot could respond to questions by instructing it to answer questions in the Google Sheets.

4.5 Writing Instructions In Natural Language

Overall, writing instructions for the blocks was a challenge. P5 (29/F) highlighted the challenge with using GPT, stating, *"This is the challenge with GPT in the first place, like providing enough information, but not too much, like you are guiding the exact answer you are getting."* P3 (22/F) also expressed uncertainty about instruction writing styles. She thought that if she wrote *"ask about the day"* as an instruction, the robot would respond with *"ask about the day"* rather than engaging the user in a conversation (e.g., "*How is your day?"*). Similarly, P2 (21/M) was uncertain about using first-person or second-person tenses in the instructions. Several participants also suggested providing more creative instruction examples.

4.6 Unexpected Surprises

Some participants were surprised by the outcomes compared to their expectations. P2 (21/M) did not expect his program to be capable of having a conversation: *"Obviously like this is super dependent on what you give it, but even the fact that it can have those conversations is pretty cool."* P4 (29/M) initially thought the task was going to be uninteresting but afterward felt that the interface let him focus on being creative: *"I didn't think it would be as fun and interesting, due to the perception of coding and it was an excel file. It seemed too organized and lame. But once you get into the system, the system is very easy to use. Like thinking about the possibilities you can create with the conversations, made it fun."*

5 DISCUSSION

Robot programming has primarily been inaccessible to non-experts without programming experience. Here, we developed an interface that enables non-expert individuals with connections to older adults to easily create the logical flow of programming a social robot for older adults, allowing them to focus on using natural language to script the robot's responses. The results of our proof-of-concept study show that individuals without a background in robot programming could successfully create a basic program involving an introduction and two questions with minimal guidance. Our high level of agreement on system usability and low task load findings imply that participants found the user interface easy to use.

The primary challenge faced by users was writing the instructions in natural language, which may be unfamiliar to prospective users. Offering a variety of creative examples for instructions can showcase the versatility of designing instructions in natural language. Additionally, users presumed that the framework had more advanced functionality, such as automatically handling a user's request for the robot to repeat itself. To address this, we could consider implementing additional backend functionality to detect and handle requests for repetition automatically.

A limitation of our programming interface is that it is not Turing-complete: it is not as powerful as a general programming language such as Python. For example, participants in this study were limited to programs with a maximum of 12 blocks and could not incorporate loops. A more dynamic version with multiple-state machines is used for complex activities in the lab, but this proposed framework allows users to begin with a simple interface for shorter interactions.

Other future work includes implementing participants' suggestions, conducting an iterative study to enhance the user experience, and assessing users' ability to create specific robot programs for interacting with older adults.

6 ACKNOWLEDGMENTS

We thank our participants and the anonymous reviewers for their valuable feedback. This work was supported by Toyota Research Institute.

REFERENCES

[1] Gopika Ajaykumar, Maureen Steele, and Chien-Ming Huang. 2021. A Survey on End-User Robot Programming. *ACM Comput. Surv.* 54, 8, Article 164 (oct 2021), 36 pages. https://doi.org/10.1145/3466819

[2] Francesca Bertacchini, Francesco Demarco, Carmelo Scuro, Pietro Pantano, and Eleonora Bilotta. 2023. A social robot connected with chatGPT to improve cognitive functioning in ASD subjects. *Frontiers in Psychology* 14 (2023). https://doi.org/10.3389/fpsyg.2023.1232177

[3] Erik Billing, Julia Rosén, and Maurice Lamb. 2023. Language Models for Human-Robot Interaction *(HRI '23)*. Association for Computing Machinery, New York, NY, USA, 905–906. https://doi.org/10.1145/3568294.3580040

[4] John Brooke. 1996. SUS: A "quick and dirty'usability. *Usability evaluation in industry* 189, 3 (1996), 189–194.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[6] Nina Buchina, Sherin Kamel, and Emilia Barakova. 2016. Design and evaluation of an end-user friendly tool for robot programming. 185–191. https://doi.org/10.1109/ROMAN.2016.7745109

[7] Neeraj Cherakara, Finny Varghese, Sheena Shabana, Nivan Nelson, Abhiram Karukayil, Rohith Kulothungan, Mohammed Afil Farhan, Birthe Nesset, Meriam Moujahid, Tanvi Dinkar, Verena Rieser, and Oliver Lemon. 2023. FurChat: An Embodied Conversational Agent using LLMs, Combining Open and Closed-Domain Dialogue with Facial Expressions. arXiv:2308.15214 [cs.CL]

[8] Enrique Coronado, Fulvio Mastrogiovanni, and Gentiane Venture. 2017. Node Primitives: an open end-user programming platform for social robots. arXiv:1709.08363 [cs.RO]

[9] Tao Dong and Luke Church. [n. d.]. Back to the future: What do historical perspectives on programming environments tell us about LLMs? ([n. d.]).

[10] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[11] Suzanne Hutson, Soo Ling Lim, Peter J Bentley, Nadia Bianchi-Berthouze, and Ann Bowling. 2011. Investigating the suitability of social robots for the wellbeing of the elderly. In *International conference on affective computing and intelligent interaction*. Springer, 578–587.

[12] Weslie Khoo, Long-Jing Hsu, Kyrie Jig Amon, Pranav Vijay Chakilam, Wei-Chu Chen, Zachary Kaufman, Agness Lungu, Hiroki Sato, Erin Seliger, Manasi Swaminathan, et al. 2023. Spill the tea: When robot conversation agents support well-being for older adults. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*. 178–182.

[13] S. Li, K. Milligan, P. Blythe, Y. Zhang, S. Edwards, N. Palmarini, L. Corner, Y. Ji, F. Zhang, and A. Namdeo. 2023. Exploring the role of human-following robots in supporting the mobility and wellbeing of older people. *Sci Rep* 13, 1 (Apr 2023), 6512.

[14] Maria R. Lima, Maitreyee Wairagkar, Manish Gupta, Ferdinando Rodriguez y Baena, Payam Barnaghi, David J. Sharp, and Ravi Vaidyanathan. 2022. Conversational Affective Social Robots for Ageing and Dementia Support. *IEEE Transactions on Cognitive and Developmental Systems* 14, 4 (2022), 1378–1397. https://doi.org/10.1109/TCDS.2021.3115228

[15] Y. C. Lin, J. Fan, J. A. Tate, N. Sarkar, and L. C. Mion. 2022. Use of robots to encourage social engagement between older adults. *Geriatr Nurs* 43 (2022), 97–103.

[16] LuxAI. 2023. LuxAI - Award winning social robots for autism and special needs education. https://luxai.com Accessed Sept. 27, 2023.

[17] Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-Shot Bot: Prompt-Based Learning for Dialogue Systems. *CoRR* abs/2110.08118 (2021). arXiv:2110.08118 https://arxiv.org/abs/2110.08118

[18] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356 [eess.AS]

[19] Natasha Randall, Swapna Joshi, Waki Kamino, Long-Jing Hsu, Abhijeet Agnihotri, Grace Li, Donald Williamson, Kate Tsui, and Selma Šabanović. 2022. Finding ikigai: How robots can support meaning in later life. *Frontiers in Robotics and AI* 9 (2022).

[20] Partha Pratim Ray. 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems* 3 (2023), 121–154. https://doi.org/10.1016/j.iotcps.2023.04.003

[21] Sanne Van Waveren, Elizabeth J Carter, Oscar Örnberg, and Iolanda Leite. 2021. Exploring non-expert robot programming through crowdsourcing. *Frontiers in Robotics and AI* 8 (2021), 646002.

[22] Ryuji Yamazaki, Hiroko Kase, Shuichi Nishio, and Hiroshi Ishiguro. 2019. A conversational robotic approach to dementia symptoms: Measuring its effect on older adults. In *Proceedings of the 7th International Conference on Human-Agent Interaction*. 110–117.

[23] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. arXiv:2304.13712 [cs.CL]

[24] Ceng Zhang, Junxin Chen, Jiatong Li, Yanhong Peng, and Zebing Mao. 2023. Large language models for human–robot interaction: A review. *Biomimetic Intelligence and Robotics* 3, 4 (2023), 100131. https://doi.org/10.1016/j.birob.2023.100131

[25] Tamara Zubatiy, Kayci L Vickers, Niharika Mathur, and Elizabeth D Mynatt. 2021. Empowering dyads of older adults with mild cognitive impairment and their care partners using conversational agents. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.