Crowdsourcing Task Traces for Service Robotics

David Porfirio
NRC Postdoctoral Research Associate
Naval Research Laboratory
Washington, DC, United States
david.porfirio.ctr@nrl.navy.mil

Allison Sauppé
University of Wisconsin–La Crosse
La Crosse, Wisconsin, United States
asauppe@uwlax.edu

Maya Cakmak
University of Washington
Seattle, Washington, United States
mcakmak@cs.washington.edu

Aws Albarghouthi
University of Wisconsin–Madison
Madison, Wisconsin, United States
aws@cs.wisc.edu

Bilge Mutlu
University of Wisconsin–Madison
Madison, Wisconsin, United States
bilge@cs.wisc.edu

ABSTRACT

Demonstration is an effective end-user development paradigm for teaching robots how to perform new tasks. In this paper, we posit that demonstration is useful not only as a teaching tool, but also as a way to understand and assist end-user developers in thinking about a task at hand. As a first step toward gaining this understanding, we constructed a lightweight web interface to crowdsource step-by-step instructions of common household tasks, leveraging the imaginations and past experiences of potential end-user developers. As evidence of the utility of our interface, we deployed the interface on Amazon Mechanical Turk and collected 207 task traces that span 18 different task categories. We describe our vision for how these task traces can be operationalized as task models within end-user development tools and provide a roadmap for future work.

CCS CONCEPTS

• **Information systems** → **Crowdsourcing**; • **Computer systems organization** → **Robotics**; • **Human-centered computing** → *User interface design*.

KEYWORDS

service robotics, crowdsourcing, end-user development

ACM Reference Format:
David Porfirio, Allison Sauppé, Maya Cakmak, Aws Albarghouthi, and Bilge Mutlu. 2023. Crowdsourcing Task Traces for Service Robotics. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI '23 Companion), March 13–16, 2023, Stockholm, Sweden.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3568294.3580112

1 INTRODUCTION

End-user developers who script personalized service robot applications face numerous challenges related to the unrestricted environments these robots often traverse, the social nuances that many of these robots must navigate, and a lack of technical expertise or appropriate development systems and tools required to address these complexities. To address these challenges, we envision end-user developer tools (such as [4, 8, 9]) using built-in task models that capture the high-level flow of common household tasks to transfer task knowledge to end-user developers. To illustrate our vision, consider an end-user developer who wishes to create a reusable, personalized task script to retrieve mail from the mailbox. Using a spoken-language interface as an example, if the end user specifies "Get the mail," the interface should leverage a *fetch* model to propose a plausible next step in the task—bring the mail to a convenient location for the end user to access later.

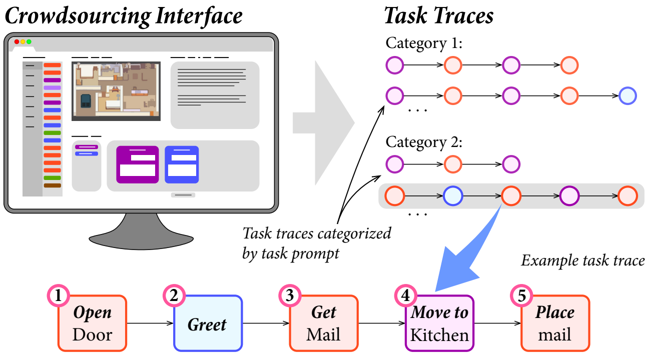


Figure 1: Our crowdsourcing interface collects individual task traces, which are organized by task category.

As a first step toward realizing our vision of transferring task knowledge to end-user developers, we (1) constructed a web interface for crowdsourcing demonstrations, or *traces*, of the step-by-step flow of common service tasks and (2) crowdsourced a preliminary dataset of 207 traces grouped within 18 separate task categories. In designing our crowdsourcing approach, our requirements were threefold. First, traces should not be tied to any particular context. As such, our trace collection interface, shown in Figure 2, presents crowdworkers with task prompts that contain minimal contextual details and encourages crowdworkers to rely heavily on their own imaginations and past experiences. Second, the collection of traces should be efficient and scalable to any imaginable service task in the home or workplace. Therefore, crowdworkers using our collection interface need only to designate *what* critical steps in a task are performed rather than *how* they are performed. Finally, traces should capture the different ways that end-user developers

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
HRI '23 Companion, March 13–16, 2023, Stockholm, Sweden

ACM ISBN 978-1-4503-9970-8/23/03…$15.00
https://doi.org/10.1145/3568294.3580112

might personalize tasks rather than the ground truth for how a robot *should* perform these tasks.

Our contributions are shown in Figure 1 and include (1) the design of an interface to collect discrete, decontextualized, and personalized task traces and (2) a dataset that we collected by deploying this interface on Amazon Mechanical Turk (MTurk).[1]

2 RELATED WORK

Our work draws from existing interfaces for simulating and demonstrating robot tasks and prior work with datasets and models that capture human activity and human-robot interaction.

2.1 Robot Simulation & Demonstration Tools

Many existing data collection and simulation interfaces for robotics have realistic environments and physics engines, such as *Habitat 2.0* [25] and *iGibson 2.0* [12]. *iGibson 2.0* additionally maps simulations to discretized, logical states that can be useful for programming or planning tasks. The *SEAN 2.0* simulator represents a further step in simulation in its ability to model the behavior of pedestrians in a scene [26]. Social behaviors are similarly present within the online game interface proposed by Chernova et al. [5], in which similar to our collection interface, interaction data is crowdsourced. An additional close match to our own needs is the *VirtualHome* simulator in its ability to crowdsource discretized, step-by-step demonstrations of tasks and social activities [17]. However, we require crowdworkers to rely on their own past experiences or imaginations rather than being provided with a realistic simulated context, thus ruling out many modern simulators such as *VirtualHome*.

2.2 Task Datasets & Models

Prior work has produced datasets and models that capture demonstrations of common household and workplace tasks, in addition to human-human and human-robot interactions. While much previous work has focused on collecting data to characterize rich multimodal sensing and behaviors [*e.g.,* 2, 10, 13], we focus instead on data that captures the discrete ordering of events in a task or interaction.

Often, such data arises from studying human behavior in the laboratory. From observing eight interaction dyads participate in five different scenarios—conversation, collaboration, instruction, interviewing, and storytelling—Sauppé and Mutlu [18] extracted models of interaction patterns in a vein similar to Kahn et al. [11]. Using the same observations, Sauppé [19] describes larger-scale models that capture the overall flow of each scenario. These laboratory-generated datasets encompass a small set of interaction scenarios, however, while we require data spanning a wider range of tasks.

Datasets also arise from outside of the laboratory, such as the *Loqui* dataset of transcribed human-human conversational interactions [15]. Also obtained in the wild, the *ARAS* [1] and *Orange4Home* datasets [6] characterize daily human activity using passive sensor data. While there is immense potential for human task and activity models to arise from this prior work, these datasets are limited in scope or capture human activity at too low of detail for characterizing individual tasks. Recent advances in large-scale in-the-wild datasets, in contrast, have proven effective in training robots to perform novel tasks [3]. However, our goal is not to transfer task skills directly to a robot; rather, we wish to capture how end-user developers imagine themselves specifying a task.

Various datasets have also been collected through simulation, such as the *ALFRED* dataset that consists of automatically-generated task demonstrations [20]. *VirtualHome* has also been used to collect demonstrations of a wide range of household tasks and social interactions [17], which in contrast to *ALFRED*, are human-generated. Existing datasets generated in simulation, however, suffer from the same drawbacks discussed in §2.1, namely being influenced by contextual characteristics enforced by the simulator.

3 TRACE COLLECTION

In this section, we describe our crowdsourcing interface, our procedure for collecting task traces, and the results of data collection.[2,3]

3.1 Collection Interface

Figure 2 depicts the crowdsourcing interface that we deployed on MTurk to collect task traces. Within the interface, the components "Prompt" (Figure 2, top right) and "Layout of Home" (Figure 2, top center) describe a category of household tasks or social scenarios that crowdworkers should imagine themselves completing. The prompt provides a textual description of the scenario, while the layout is intended to assist crowdworkers in understanding and situating themselves within the prompt. The layout is minimally interactive, allowing crowdworkers to hover over it and receive information about potentially relevant rooms or entities within the home. The prompt and layout are purposefully low in detail in order to stimulate imagination and the recollection of past experiences to fill in missing task details. §3.2 provides a list of the 18 prompt categories we used within our trace collection procedure.

Crowdworkers respond to a prompt by dragging task *steps* from the "Toolbox" component and dropping them into the "Task Timeline" to create a task trace. Table 1 defines the 17 parameterizable steps available in the interface (Figure 2, left), which are intended to map to robot skills. The interface provides descriptions of each step as tooltips.

To instantiate a step, crowdworkers click on it or drag it from the toolbox to the timeline, after which it becomes available to be parameterized. Figure 2 (bottom) displays an instantiated **approach** step, in which the *person* parameter is parameterized with the *Guest 1* argument. Therefore, the first step in the trace is **approach:** *Guest 1.* To facilitate open-endedness in addressing the prompts, parameterization is free response. We provided an additional free response text box at the bottom of each instantiated step to allow crowdworkers to provide additional detail or justify a particular step.

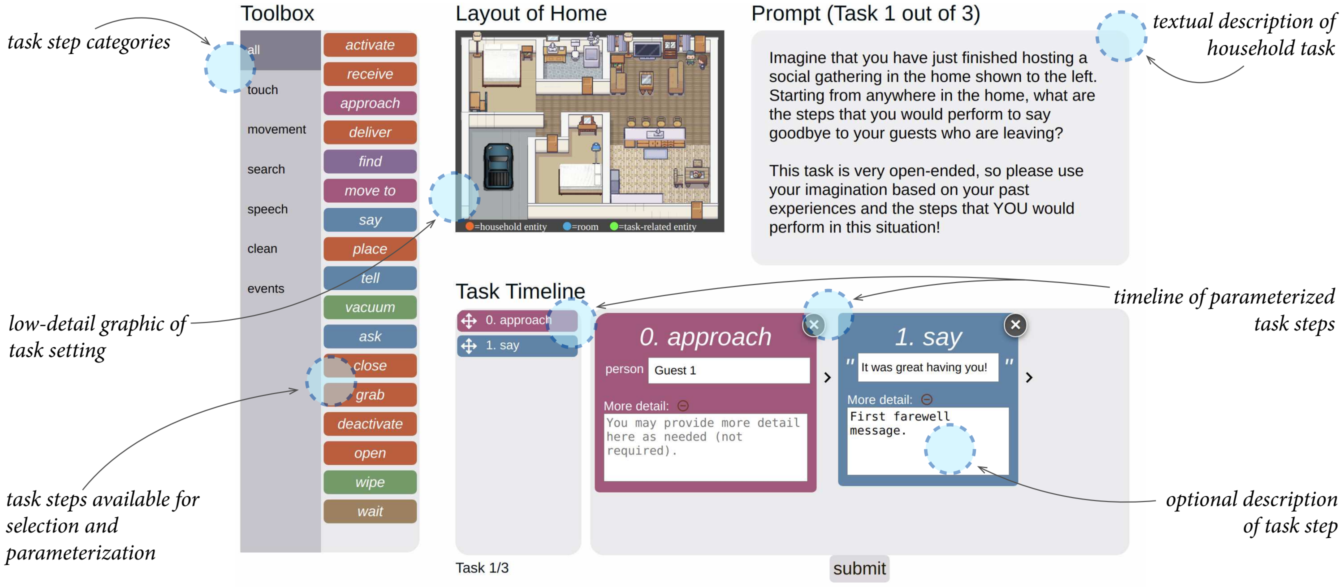
3.2 Collection Procedure

We conducted an IRB-approved study in which 105 MTurk crowdworkers (*Turkers*) used our collection interface. To participate, Turkers needed IP addresses geographically within the United States and a >95% task approval rate. In providing consent to participate, Turkers were informed that their data would be publicly shared and that their responses were subject to approval by the research team.

[1] https://www.mturk.com/

[2] Portions of §3 were presented in Chapter 7 of the first author's Ph.D. thesis [16].

[3] Our study materials, code for the web interface, and resulting dataset can be found at https://osf.io/jt9hr



Imagine that you have just finished hosting a social gathering in the home shown to the left. Starting from anywhere in the home, what are the steps that you would perform to say goodbye to your guests who are leaving?

This task is very open-ended, so please use your imagination based on your past experiences and the steps that YOU would perform in this situation!

person: Guest 1

" It was great having you! "

More detail: First farewell message.

task step categories

low-detail graphic of task setting

task steps available for selection and parameterization

textual description of household task

timeline of parameterized task steps

optional description of task step

Figure 2: The web interface that we created to collect task traces. The Layout of Home pane uses graphics from LimeZu.[4]

Turkers whose work we approved were paid $2.67 for an expected completion time of 20 minutes.

After giving their consent to participate, Turkers were directed to a tutorial web page that described via text and a three-minute YouTube video tutorial how to use the interface and the criteria for their responses to be approved. Our approval criteria were such that (1) Turkers could not provide traces with only one step, and (2) traces must address the provided task prompts. We encouraged Turkers to use the optional free response description text boxes under each step in the timeline to justify their work (*i.e.,* if a response outwardly seemed irrelevant to the prompt). Generally, we did not discard traces from Turkers who provided justifications. If fewer than two traces provided by a Turker failed to meet the acceptance criteria outlined on the tutorial page, we discarded all of the Turker's traces.

`move to:` *target*	—	move to a *target*
`find:` *target*	—	search for a *target*
`grab:` *item*	—	grab an *item*
`open:` *container*	—	open a *container*
`close:` *container*	—	close a *container*
`deliver:` *item, target*	—	bring an *item* to a *target*
`receive:` *item*	—	receive an *item* from someone
`place:` *item, container*	—	place an *item* in a *container*
`approach:` *person*	—	approach a *person*
`say:` *exact-speech*	—	say the *exact speech* as specified
`tell:` *story*	—	tell a *story*
`ask:` *exact-speech*	—	ask a question using *exact speech*
`activate:` *device*	—	turn a *device* on
`deactivate:` *device*	—	turn a *device* off
`vacuum:` *room*	—	clean a *room* by vacuuming it
`wipe:` *surface*	—	clean a *surface* by wiping it
`wait`	—	wait for something to happen

Table 1: The parameterizable steps available within our crowdsourcing interface.

We used the YouTube IFrame Player API[5] to ensure that Turkers watched the video before proceeding to the collection interface. Once allowed to proceed, each was asked to respond to three separate task prompts. An example *Mail* prompt is as follows:

> *Imagine that you live in the home shown to the left, and the mail has just arrived through a slot in the front door. Before opening any of the letters or packages and starting from anywhere in the home, what steps would you take to fetch the mail?*

Each prompt then ends with the statement *"This task is very open-ended, so please use your imagination based on your past experiences and the steps that YOU would perform in this situation!"*

In addition to the *Mail* prompt, an additional 17 prompts captured tasks within the following categories: *Greeting*, *Farewell*, *Groceries*, *Storytelling*, *Alarm*, *Announcement*, *Vacuum*, *Answer Door*, *Turn on Lights*, *Delivery*, *Ask About Day*, *Phone Call*, *Patrol*, *Find*, *Dust*, *Declutter*, and *Answer Question*. In designing each prompt, we aimed to include a spectrum of both social and non-social tasks and intended for the prompts to be general enough to allow crowdworkers immense freedom in how they chose to respond.

3.3 Collection Results

Of the first 69 Turkers, we rejected 12 (82.6% approval rate). Of the remaining 36, who participated at later dates than the initial 69, we noticed a substantial increase in spam responses and only approved 13 (36.1% approval rate). We observed no difference in the quality of approved responses after the increase in spam. Of

[4] https://limezu.itch.io/

[5] https://developers.google.com/youtube/iframe_api_reference

the Turkers whose work we include in the dataset, three provided a single trace that did not meet our approval criteria, so we discarded these individual traces and kept the remainder of their work.

The final dataset includes 207 traces sourced from 70 Turkers. On average, the number of traces collected per task category is 11.5 (min = 10 traces, max = 16 traces). On average, traces contained 6.23 steps (min = 2 steps, max = 23 steps). Turkers appeared attentive and thoughtful when using the interface—62 Turkers used the free response boxes on their instantiated steps to provide additional detail, with the total number of descriptions being 706 (a rate of 0.55 descriptions per step, with 1289 total steps provided).

Participants were additionally asked for feedback on the interface after they finished responding to all three prompts, and their positive feedback further indicates high engagement. Examples of feedback include the following:

> *This was one of the more unique tasks I've done on mturk. It was very interesting.*
>
> *I would love to do more of these! It reminds me of solving problems using pseudocode!*
>
> *I enjoyed it, it felt like a roleplaying game.*

The feedback also indicates that some Turkers were uncertain as to whether they completed the task correctly:

> *I believe I did well on this task but would appreciate any feedback.*
>
> *I would have liked to receive feedback during the exercise to know if I was hitting the mark and accomplishing the goal of the study. I tried my best.*

Other interesting phenomena occurred during the collection of traces. 61.4% of Turkers used the **wait** step, which was often accompanied by free response descriptions of external events that must occur in order for the task to proceed (*e.g.,* waiting for a verbal response). Additionally, many traces explicitly resolve preconditions that an autonomous robot could likely resolve by itself. In the step **wipe:** *table*, for example, a robot is likely aware of the necessary precondition of being in possession of a duster or cloth, and if it does not have one, of the need to fetch one. Turkers were not aware that their traces were intended to be used for service robots and often explicitly included these preconditions in their traces.

4 DISCUSSION

In this work, we provide a lightweight approach for crowdsourcing task traces with the goal of transferring knowledge from the traces to end-user developers who script tasks for service robots. As developers specify the high-level steps for a robot to perform (*e.g.,* a command to put a bag of groceries in the kitchen) a model of the task at hand can suggest ways that the task specification can be further personalized, such as by inferring edits to the task flow or by prompting clarification on unstated task structure. For example, a developer tool may suggest a foreach loop to an end user specifying a *Groceries* task.

In transferring task knowledge to end-user developers, we are inspired by previous work in which program hints provided by a developer serve as input to an automated program synthesizer for completion [*e.g.,* 21–24]. In our case, we envision a human-in-the-loop pipeline in which end-user developers provide a set of task steps as hints, and a developer tool uses a task model constructed from task traces to suggest where additional steps, loops, or branch points might be needed. This vision is closely aligned with prior work in human-robot interaction, in which *templates* have been used as pre-existing generic, reusable program specifications to be selected and instantiated by end users [7].

To achieve our vision, we aim to create task models from individual traces (such as in [14]) or multiple task traces. A naïve approach could treat individual traces as models themselves and compute a "diff" between the hint and the trace to find missing steps omitted by the end user. In a more sophisticated approach, multiple traces under the same category could be combined into a probabilistic model, such as a Markov chain, that could be used to compute the probability of a particular step, loop, or branch point being present in the task. If there is uncertainty in the task hint, such as if the end user provides an ambiguous or incomplete step (*e.g.,* the spoken language utterance "that goes over there"), combined traces may also serve as hidden Markov models, in which the current step of the task being specified by the end user must be inferred.

Our dataset currently limits our ability to pursue the more sophisticated, probabilistic approach, as we would likely need to collect many more traces and rigorously post-process these traces to remove noise. Our dataset is further limited in the number of task categories, currently 18, within which we sourced traces. Future work must therefore involve (1) collecting more task traces per task category in order to sufficiently build models that represent the different ways that the same type of task can be performed, and (2) modifying our interface to streamline the collection of more traces.

Although our interface is lightweight and can scale to new task categories, it too has limitations that hinder data collection. First, in order to introduce a new task category, a researcher must manually construct a new prompt, potentially injecting their own biases. Future work should enable task categories to naturally emerge from the collection of crowdworkers' daily routines. Additionally, at present time, individual traces provided by crowdworkers must be manually screened by a researcher. Although we believe that our trace acceptance criteria are highly objective, future work must strive toward more systematic criteria, preferably automated so as to remove any possible researcher bias or error.

5 CONCLUSION

We present a web interface for collecting traces of service robot tasks and a small dataset collected from the deployment of this interface on Amazon Mechanical Turk. We describe our vision of aggregating task traces to create task models that developer tools can use to assist end users in scripting personalized service robot tasks. Our dataset demonstrates that the interface is scalable to a large number of tasks and is easy for demonstrators to use.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) award 1925043 and a University of Wisconsin–Madison Letters & Science Community of Graduate Research Scholars Fellowship. This work was conducted while David Porfirio was affiliated with the University of Wisconsin–Madison and finalized while an NRC Postdoctoral Research Associate at the Naval Research Laboratory.

REFERENCES

[1] Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. 2013. ARAS Human Activity Datasets in Multiple Homes with Multiple Residents. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare* (Venice, Italy) *(PervasiveHealth '13)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 232–235. https://doi.org/10.4108/icst.pervasivehealth.2013.252120

[2] Atef Ben-Youssef, Chloé Clavel, Slim Essid, Miriam Bilac, Marine Chamoux, and Angelica Lim. 2017. UE-HRI: A New Dataset for the Study of User Engagement in Spontaneous Human-Robot Interactions. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction* (Glasgow, UK) *(ICMI '17)*. Association for Computing Machinery, New York, NY, USA, 464–472. https://doi.org/10.1145/3136755.3136814

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. 2022. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv preprint arXiv:2212.06817* (2022).

[4] Yuanzhi Cao, Zhuangying Xu, Fan Li, Wentao Zhong, Ke Huo, and Karthik Ramani. 2019. V.Ra: An In-Situ Visual Authoring System for Robot-IoT Task Planning with Augmented Reality. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) *(DIS '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. https://doi.org/10.1145/3322276.3322278

[5] Sonia Chernova, Nick DePalma, Elisabeth Morant, and Cynthia Breazeal. 2011. Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *2011 RO-MAN*. 21–26. https://doi.org/10.1109/ROMAN.2011.6005284

[6] Julien Cumin, Grégoire Lefebvre, Fano Ramparany, and James L. Crowley. 2017. A Dataset of Routine Daily Activities in an Instrumented Home. In *Ubiquitous Computing and Ambient Intelligence*, Sergio F. Ochoa, Pritpal Singh, and José Bravo (Eds.). Springer International Publishing, Cham, 413–425. https://doi.org/10.1007/978-3-319-67585-5_43

[7] Paola Ferrarelli, María T. Lázaro, and Luca Iocchi. 2018. Design of Robot Teaching Assistants Through Multi-modal Human-Robot Interactions. In *Robotics in Education*, Wilfried Lepuschitz, Munir Merdan, Gottfried Koppensteiner, Richard Balogh, and David Obdržálek (Eds.). Springer International Publishing, Cham, 274–286. https://doi.org/10.1007/978-3-319-62875-2_25

[8] Dylan F. Glas, Takayuki Kanda, and Hiroshi Ishiguro. 2016. Human-Robot Interaction Design Using Interaction Composer: Eight Years of Lessons Learned. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction* (Christchurch, New Zealand) *(HRI '16)*. IEEE Press, 303–310. https://doi.org/10.1109/HRI.2016.7451766

[9] Justin Huang and Maya Cakmak. 2017. Code3: A System for End-to-End Programming of Mobile Manipulator Robots for Novices and Experts. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (Vienna, Austria) *(HRI '17)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/2909824.3020215

[10] Dinesh Babu Jayagopi, Samira Sheiki, David Klotz, Johannes Wienke, Jean-Marc Odobez, Sebastien Wrede, Vasil Khalidov, Laurent Nyugen, Britta Wrede, and Daniel Gatica-Perez. 2013. The vernissage corpus: A conversational Human-Robot-Interaction dataset. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 149–150. https://doi.org/10.1109/HRI.2013.6483545

[11] Peter H. Kahn, Nathan G. Freier, Takayuki Kanda, Hiroshi Ishiguro, Jolina H. Ruckert, Rachel L. Severson, and Shaun K. Kane. 2008. Design Patterns for Sociality in Human-Robot Interaction. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction* (Amsterdam, The Netherlands) *(HRI '08)*. Association for Computing Machinery, New York, NY, USA, 97–104. https://doi.org/10.1145/1349822.1349836

[12] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. 2022. iGibson 2.0: Object-Centric Simulation for Robot Learning of Everyday Household Tasks. In *Proceedings of the 5th Conference on Robot Learning (Proceedings of Machine Learning Research, Vol. 164)*, Aleksandra Faust, David Hsu, and Gerhard Neumann (Eds.). PMLR, 455–465. https://proceedings.mlr.press/v164/li22b.html

[13] Yasser Mohammad, Yong Xu, Kenichi Matsumura, and Toyoaki Nishida. 2008. The H3R Explanation Corpus human-human and base human-robot interaction dataset. In *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 201–206. https://doi.org/10.1109/ISSNIP.2008.4761987

[14] Anahita Mohseni-Kabir, Charles Rich, Sonia Chernova, Candace L. Sidner, and Daniel Miller. 2015. Interactive Hierarchical Task Learning from a Single Demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (Portland, Oregon, USA) *(HRI '15)*. Association for Computing Machinery, New York, NY, USA, 205–212. https://doi.org/10.1145/2696454.2696474

[15] Rebecca Passonneau and Evaneet Sachar. 2014. Loqui Human-Human Dialogue Corpus (Transcriptions and Annotations). (2014). https://doi.org/10.7916/D82R3PW9 Available from https://academiccommons.columbia.edu/doi/10.7916/D82R3PW9.

[16] David J Porfirio. 2022. *Authoring Social Interactions Between Humans and Robots*. Ph. D. Dissertation. The University of Wisconsin–Madison.

[17] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. VirtualHome: Simulating Household Activities Via Programs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8494–8502. https://doi.org/10.1109/CVPR.2018.00886

[18] Allison Sauppé and Bilge Mutlu. 2014. Design Patterns for Exploring and Prototyping Human-Robot Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1439–1448. https://doi.org/10.1145/2556288.2557057

[19] Allison V Sauppé. 2015. *Designing effective communication strategies for human-robot collaboration.* Ph. D. Dissertation. The University of Wisconsin-Madison.

[20] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10737–10746. https://doi.org/10.1109/CVPR42600.2020.01075

[21] Armando Solar-Lezama. 2008. *Program synthesis by sketching.* Ph. D. Dissertation. University of California, Berkeley.

[22] Armando Solar-Lezama. 2009. The Sketching Approach to Program Synthesis. In *Proceedings of the 7th Asian Symposium on Programming Languages and Systems* (Seoul, Korea) *(APLAS '09)*. Springer-Verlag, Berlin, Heidelberg, 4–13. https://doi.org/10.1007/978-3-642-10672-9_3

[23] Armando Solar-Lezama. 2013. Program Sketching. *International Journal on Software Tools for Technology Transfer* 15, 5–6 (oct 2013), 475–495. https://doi.org/10.1007/s10009-012-0249-7

[24] Saurabh Srivastava, Sumit Gulwani, and Jeffrey S. Foster. 2013. Template-Based Program Verification and Program Synthesis. *International Journal on Software Tools for Technology Transfer* 15, 5–6 (oct 2013), 497–518. https://doi.org/10.1007/s10009-012-0223-4

[25] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimír Vondruš, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. 2021. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 251–266. https://proceedings.neurips.cc/paper/2021/file/021bbc7ee20b71134d53e20206bd6feb-Paper.pdf

[26] Nathan Tsoi, Alec Xiang, Peter Yu, Samuel S. Sohn, Greg Schwartz, Subashri Ramesh, Mohamed Hussein, Anjali W. Gupta, Mubbasir Kapadia, and Marynel Vázquez. 2022. SEAN 2.0: Formalizing and Generating Social Situations for Robot Navigation. *IEEE Robotics and Automation Letters* 7, 4 (2022), 11047–11054. https://doi.org/10.1109/LRA.2022.3196783