# Towards a Natural Language Programming Interface for Smart Homes

**Meghan Clark**
University of Michigan
EECS Department
Ann Arbor, MI 48109, USA
mclarkk@umich.edu

**Prabal Dutta**
University of Michigan
EECS Department
Ann Arbor, MI 48109, USA
prabal@umich.edu

**Mark W. Newman**
University of Michigan
School of Information
Ann Arbor, MI 48109, USA
mwnewman@umich.edu


*UbiComp/ISWC '16* Adjunct, September 12–16, 2016, Heidelberg, Germany
ACM 978-1-4503-4462-3/16/09.
http://dx.doi.org/10.1145/2968219.2971443

## Abstract
Recently several major industry players have introduced smart home personal assistants to the Internet of Things market. However, these products are often little more than voice interfaces to device control apps. More focus is needed on supporting the high-level applications and goals that end users express in natural language. In this preliminary work, we collect and analyze over 1,600 utterances from non-technical end users encompassing a variety of smart home interactions. We find that given the right context, end users voluntarily use computational language in response to open-ended prompts. From these commands and queries, we extract the computational structures and the entities, actions, and conditions that end users employ when interacting with the home. Our preliminary results suggest a path forward for end-to-end architectures that automatically map natural language utterances to executable smart home programs. We hope to help bring natural language interfaces for the home into alignment with end users' mental models.

## Author Keywords
Smart Home; Internet of Things; Natural Language

## ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

## Introduction

Over the last few years, several personal assistants with voice interfaces have entered the smart home market [1, 3, 4]. However, so far these platforms have supported direct device actions and if-this-then-that (trigger-action) programs with a limited set of triggering conditions and subsequent actions. Are we providing the right abstractions and programming primitives to capture end users' applications?

Previous work has examined natural language for patterns in how end users specify domestic computing applications [5, 6]. However, these studies have constrained the input from end users and emphasized certain kinds of interactions with the home, specifically action-oriented directives. Prior work has also not deeply explored the different kinds of conditions under which actions take place.

Our analysis of over 1,600 survey responses reveals two different ways of expressing *queries*, which prior work has not highlighted due to its emphasis on trigger-action. We also make the case that triggers are just one kind of condition on action, and describe several other classes of conditions. Our preliminary results highlight several new roles for artificial intelligence and machine learning in smart home systems, and suggest future directions for mapping natural language utterances to executable programs.

## Study Methodology

We devised two surveys to collect data about end user programs for smart homes. At a high level, each survey described the capabilities of a hypothetical smart home and asked the respondent what they wanted the home to do. However, we found that differences in the way we presented the smart home's capabilities strongly affected the nature and structure of the responses that we received.

*System Prompt*

The first survey asked respondents to imagine that they lived in a smart home, and provided a description of the sensing and actuation capabilities of the smart home. We were concerned that the abstractions we used in our description of the smart home's capabilities would affect the mental models people used to construct applications, so we gave subjects one of two smart home descriptions with equal likelihood: 1) a device-oriented description, and 2) a data-oriented description. The device-oriented description listed example smart devices that the respondent might have available in their home broken down by sensor, actuator, or online service. The data-oriented description listed example data streams, broken down by read-only data (sensor readings), read-write data (actuator statuses), or online service data. The devices and data streams corresponded to the same set of smart home capabilities. The respondents were given a minute to look over the list, then they were asked to write for five minutes about what kinds of applications they wanted in their smart home. We received 615 extensive free-form responses about what the respondents want their smart home to do.

*AI Prompt*

In the second survey we introduced a completely loyal artificial intelligence agent (an "AI") as the intelligence behind the smart home controls. This survey first asked respondents to select a gender for their AI's voice from a randomly ordered list of male, female, and androgynous, then respondents were asked to name their AI. Having imbued their hypothetical smart home with independent yet trustworthy agency, we then gave respondents one of the two smart home capability descriptions from the first survey. The respondents were given a minute to look over the list of devices or data. Afterwards, we told respondents that their AI wanted to help them by automating their home on their

behalf. We asked them to tell the AI what it should do, and began the writing prompt with, “OK, <AI name>..." to encourage respondents to “speak" – that is, write – directly to the smart house. We received 1,054 responses.

*Subjects*
We submitted each survey to Amazon Mechanical Turk [2]. We limited respondents to the United States, the United Kingdom, Canada, Australia, and New Zealand in order to increase the proportion of native English speakers in the eligible respondent pool. Respondents were slightly more male, with 60% male and 40% female. Most respondents were young, with 45% age 25-34, 21% age 18-24, 19% age 35-44, 9% age 45-54, and 6% age 55+. However, the distribution of educational attainment was similar to that of the United States. Most respondents were not knowledgeable about computer science. 12% of respondents categorized their occupation as either computer worker or engineer, but 65% of respondents considered themselves to have “low" exposure to CS concepts and 11% had no exposure at all. 66% had never heard of the Internet of Things before.

## Preliminary Results

The qualitative difference in the responses between the first survey, which simply described the smart devices or data available in the home, and the second survey, which additionally provided an AI character, is marked. The first survey failed to elicit high-level integrated applications. Many respondents appeared to think that *they* had to be the central controller, receiving all information and deciding what to actuate themselves. In sharp contrast, the addition of an AI character in the second survey inspired end users to respond with programmatic directives or queries to open-ended prompts. This is the dataset we analyze throughout the rest of this section.

*What kinds of primitives (triggers, actions, entities, etc.) should end-user programming interfaces provide?*
We found 981 and 283 unique nouns and verbs, respectively. The top 42 nouns accounted for 50% of all noun usage, and the top 212 nouns covered 80%. The top 20 verbs accounted for 65% percent of total verb usage, and the top 85 verbs covered 90%. From this we can see that a system with a fixed set of known actions and an extensible list of known entities could handle the majority of the most common cases. Using Jaccard index calculations we also found that certain verbs were closely associated with certain nouns, while other verbs were more general, which could aid in the design of device-level interfaces.

To study the computational structure of these utterances, we took a set of responses that contained the top 100 most common nouns and verbs and hand-crafted a grammar that could express all the utterances in a canonical form. The structure of the resulting grammar revealed certain patterns. Utterances fell under one of several categories: Direct actions (“Turn on the lights"), conditional actions (“Turn on the lights when I come home"), queries in the form of a question (“How much do I weigh?"), and queries in the form of a direct action (“Tell me how much I weigh"). These different forms of queries have not been mentioned in previous work. Additionally, direct actions break down further into a variety of patterns, most of which fall under one of the following: actions involving objects, actions involving knowable things, and actions involving living entities.

Our grammar method identified a number of previously unknown conditions. The list of conditions we found under which an action can occur is: 1) whenever an event happens, 2) while a fact is true, 3) before an event or time, 3) after an event or time, 4) until an event, fact, or time, 5) unless an event happens or a fact is true, 6) to achieve a goal

("in order to", "so that"), 7) only when the nouns involved match a certain attribute or relation, 8) location, 9) time, 10) data source (“based on the outdoor motion sensors"). A number of these conditions are difficult to capture using the event-based “triggers" from trigger-action programming.

*How can we take natural language utterances and turn them into executable programs?*
The structure of our grammar revealed that most utterances correspond to one of several *speech acts* from linguistics. We can use a dependency parse tree to classify the likely act intended by an utterance, and extract the relevant information about entities, attributes, and conditions from the prepositional phrases. This will allow us to convert any natural language utterance to a canonical representation, which we can then map to an executable program.

## Design Implications

Above we showed that end users naturally employ computational language when speaking to a smart home AI, and that a finite set of actions, entities, and speech acts can capture a large percentage of use cases. However, we also made the case that existing smart home voice interfaces do not expose the right kinds of primitives, particularly in terms of conditions. We outlined an approach to build a better end-to-end system, which we plan to explore in future work.

One other surprising outcome of our analysis is the identification of new roles that classic AI and machine learning both can play in the IoT application space. AI or ML are required for the following: 1) conditional actions that require prediction in order to ensure that the actions are completed before future predicted events occur (“Have the coffee ready *before* I wake up"), 2) conditional actions that require an understanding of goals and how to attain them, like a classic AI planning agent (“Track my dog *so that* I know where he goes when he's outside"), 3) queries about the future (“When will my kids get home?"), which would also be enabled by the predictive abilities of machine learning, and 4) “why" queries, which are answerable given a sophisticated enough classic AI inference engine or a planning agent (“Why did the music turn on?", “Why am I so tired?").

## REFERENCES

1. Amazon.com, Inc. 2014. Amazon Echo. https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E. (2014).
2. Amazon.com, Inc. 2015. Amazon Mechanical Turk. https://www.mturk.com. (2015).
3. Google, Inc. 2016. Google Home. https://home.google.com. (2016).
4. Protonet, Inc. 2016. ZOE. https://www.indiegogo.com/projects/protonet-zoe-start-your-secure-smart-home-now. (2016).
5. Khai N. Truong, Elaine M. Huang, and Gregory D. Abowd. 2004. CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home. In *UbiComp 2004: Ubiquitous Computing*, Elizabeth D. Mynatt Nigel Davies and Itiro Siio (Eds.). Lecture Notes in Computer Science, Vol. 3205. Springer Berlin Heidelberg, 143–160. DOI: http://dx.doi.org/10.1007/978-3-540-30119-6_9
6. Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical Trigger-Action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 803–812.