

Supporting Mental Model Accuracy in Trigger-Action Programming

Justin Huang and Maya Cakmak

Computer Science & Engineering Department, University of Washington
185 Stevens Way, Seattle, Washington, USA
{jstn,mcakmak}@cs.washington.edu

ABSTRACT

Trigger-action programming is a simple programming model that enables users to create rules that automate behavior of smart homes, devices, and online services. Existing trigger-action programming systems, such as *if-this-then-that* (IFTTT), already have millions of users worldwide; however, their oversimplification limits the expressivity of the programs that can be created. While extensions of IFTTT to allow more complex programs have been proposed, previous work neglects a key distinction between different trigger types (states and events) and action types (instantaneous, extended, and sustained actions). In this paper, we systematically study the impact of these differences through two user studies that reveal: (i) inconsistencies in interpreting the behavior of trigger-action programs and (ii) errors made in creating programs with a desired behavior. Based on a characterization of these issues, we offer recommendations for improving the IFTTT interface so as to mitigate issues that arise from mental model inaccuracies.

Author Keywords

Trigger-action programming; IFTTT; smart homes

ACM Classification Keywords

H.1.2. User/Machine Systems: Human factors

INTRODUCTION

Trigger-action programming (TAP) is a simple programming model in which the user associates a *trigger* with an *action*, such that the action is automatically executed when the trigger event occurs. The most popular TAP interface is an online service called *if-this-then-that*¹ (IFTTT). IFTTT allows users to create programs that can automatically perform actions like sending alerts or changing settings of a smart home, when certain triggers occur (e.g., it starts raining, someone tags the user in a picture, the user arrives at home, *et*

¹www.ifttt.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UbiComp '15, September 7–11, 2015, Osaka, Japan
©2015 ACM. ISBN 978-1-4503-3574-4/15/09 \$15.00
DOI: <http://dx.doi.org/10.1145/2750858.2805830>

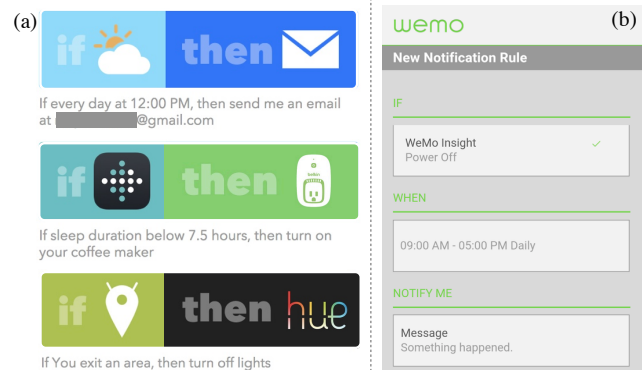


Figure 1: Trigger-action programming in existing products. (a) Example rules from IFTTT. (b) An example of programming a rule for a WeMo Insight Switch.

cetera). Fig.1(a) shows three examples of such IFTTT programs. With its increasing support for wearables, smartphone sensors, and other connected devices (e.g., Nest Thermostats, Belkin WeMo switches, or Phillips Hue lightbulbs), IFTTT has become highly relevant for ubiquitous computing. The simplicity of IFTTT has allowed millions of everyday users to create simple programs without requiring any specialized programming skills, hence addressing an important challenge in ubiquitous computing in the home [5].

Despite its widespread and diverse use, IFTTT has an important restriction that limits its expressivity. It only allows a single event to be used as triggers in programs. As a result, it does not support rules that may be relevant in the context of multiple triggers. For example, a user might want to receive notifications that the motion sensor in their home was activated while they are not at home; however, IFTTT does not allow this simple conjunction of triggers (user not at home and motion sensor activated) and hence is unable to express this rule.

Previous work on TAP has noted this limitation and proposed systems that allow conjunctions of multiple triggers [19, 4, 8]. Although their user studies have demonstrated that people are able to use multiple triggers to create complex programs with correct behavior, none of them have investigated the accuracy of the user's mental model about how exactly those programs behave. In particular, previous work does not distinguish be-

tween conceptually different trigger types and action types. This can cause ambiguities in terms of when exactly the action would be activated and whether or not certain actions would be automatically reverted. Errors caused by such misunderstandings can have serious consequences; for instance in the context of home automation, program errors could risk home security by unlocking doors at the wrong time or cause unintended energy waste by not reverting a thermostat setting.

In this paper, we first identify these distinct types of triggers (event and state triggers) and actions (instantaneous, extended, and sustained actions) that are used in existing systems or in previous research. Then, we present a study that reveals inconsistencies in *interpreting* the behavior of single-trigger and multiple-trigger programs involving these different types of triggers and actions. We follow up with another study that reveals errors made in *creating* similar programs. Finally, based on a characterization of these issues, we offer recommendations for improving the IFTTT interface so as to mitigate user mental model inaccuracies.

RELATED WORK

Previous work has established the importance of TAP in the context of smart homes, by showing that it can express most behaviors desired by potential users. In the work done by Ur et al. [19], users were asked to list smart home behaviors they wanted, and the authors found that all of the behaviors which required programming could be expressed in a trigger-action format. Similarly, Dey et al. [4] asked users to think of open-ended behaviors for smart homes, and showed that close to 80% of the described behaviors fit an if-then format.

Researchers have also developed and evaluated different TAP interfaces. Ur et al. developed an IFTTT-like interface that supported multiple triggers and multiple actions, and found that users could correctly create a given set of rules about 80% of the time [19]. Dey et al. built a visual programming system (iCAP) in which triggers and actions could be dragged onto a rule sheet [4]. Their user study evaluating this interface showed that non-programmers were able to program rules to implement a given set of behaviors. Häkkinen et al. implemented a TAP system (Context Studio) for customizing a mobile phone [8]. Zhang et al. [21] presented work on visually debugging and exploring event-condition-action rules for robot behaviors. Mackay et al. [15] studied rule creation for an email filtering system. There is also a body of work in database systems research on *production rules* [9] which enable useful actions in database systems to be automatically triggered when a condition is met.

Our work is most closely related to the first three papers mentioned above, all of which included support for conjunctions of multiple triggers. As detailed in the next section, their systems (as well as other existing systems) use different types of triggers and actions without acknowledging their distinction. Our work contributes a theoretical description of these distinct types as well as empirical findings demonstrating ambiguities due to these distinctions from two user studies in which trigger and action types are systematically varied.

Our work fits into a larger space of smart home and context-aware programming research. Jahnke et al. [10] describe several modes of programming smart homes, one of which is *connection-based* programming. In this model, devices made by several parties are connected together by explicitly interfacing with each other. One device can provide a callback to another device, and can execute actions based on events of interest. Welbourne et al. [20] describe the design of a system for designing and verifying location triggers by modeling them as finite state machines. Truong et al. [18] provide a different programming model for smart homes, through the metaphor of magnetic poetry. In this model, users arrange tokens to form sentences describing desired behaviors.

Our work also relates to the general challenge of enabling end-user programming. In previous research, end-user programming has been enabled through a variety of methods, including programming by demonstration [14], graphical interfaces for programming robots [12], [2], [1], and translating or adopting natural language [11], [17]. Researchers have also evaluated programming interfaces designed for novices [7]. More overviews of this area are provided by [6] and [16], with some challenges listed in [13].

TRIGGER AND ACTION TYPES

Trigger-action programs in IFTTT consist of a single rule that associates a trigger with an action. The wording “if this then that” suggests that they are considered as being equivalent to an if-then statement afforded in many general purpose programming languages such as Java or Python; however, their semantics are in fact equivalent to event-driven programming [3] in which an asynchronous input signal (trigger) is handled by a callback function (action). This inaccuracy in the if-then metaphor creates some ambiguities.

One ambiguity arises from the lack of distinction between two trigger types (Figure 2):

- *Events* which are instantaneous signals, versus
- *States* which are boolean conditions that can be evaluated to be true or false at any time.

Events indicate the occurrence of some change at a specific point in time. Examples of event triggers include, “the doorbell rings,” or “the temperature drops below 50° F.” State triggers indicate that some condition is currently true, lasting over a period of time. Examples of state triggers could be “it is between 3:00 - 5:00 pm,” or “it is raining.” While regular if-then rules in general-purpose programs are meaningful with state type triggers, TAP with a single trigger is mostly meaningful with event type triggers. Indeed, IFTTT avoids state-type triggers through the use of events associated with state changes, e.g., “if the temperature *drops* below” instead of “if the temperature *is* below.”

We nonetheless observe the use of state-type triggers in previous work. For example, Dey et al.’s user study involves creating rules such as “If I am sleeping, turn the stereo off” [4]. The meaning of this rule with the state-type trigger “if I am sleeping” is ambiguous—does the rule start as soon as I fall asleep, or does the rule start any time while I am asleep?

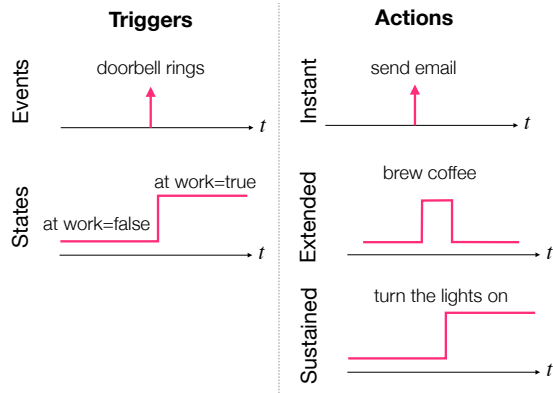


Figure 2: The different kinds of triggers (left) and actions (right) we distinguish between in this paper. The horizontal axis in each diagram represents time. A raised value indicates that the trigger or action is active at that point in time.

We hypothesize that most people would choose the former interpretation. The reason for this is because if I fall asleep and the rule did not activate immediately, then there would be a period of time during which the rule did not hold.

A similar distinction can be made with action types (Fig. 2):

- Some actions are *instantaneous* and do not change the state of the system. An example is sending an alert (email or a text message). This action can be completed within one time step and the system would be ready to send another alert at the next time step.
- Other actions are *extended* in time but they are completed within a certain, often deterministic, amount of time. An example is brewing coffee. This action can take a few minutes during which the same action cannot be executed because the actuator is busy. However, the action completes automatically and reverts back to its original state.
- In contrast, *sustained* actions involve changing the state of an actuator, such as turning the lights on/off or setting the thermostat temperature. This new state does not revert back automatically, as with extended actions that come to a conclusion.

IFTTT allows all three types of actions (see three examples in Fig.1(a)) but it does not make a distinction between them. This can be problematic particularly for sustained actions, since reverting their effects requires a separate rule and users might have false expectations that they are automatically undone, particularly when paired with a state-type trigger. For instance, in the example quoted earlier “If I am sleeping, turn the stereo off” [4] the fact that there is an end to the state trigger (i.e., I will wake up at some point) may imply that the action will also end—i.e., the stereo will turn back on—while others would say that the stereo would not turn on unless directed to by another rule.

Trigger conjunctions

In IFTTT, users can only create rules with exactly one trigger. This limits the expressivity of the rules, because multiple triggers cannot be combined. For example, suppose a user

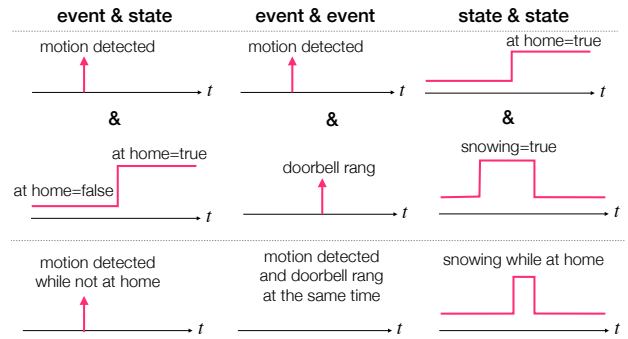


Figure 3: Conjunctions of different trigger types.

Trigger(s)	Logical interpretation
Single event	Rule activates when event occurs
Single state	Rule activates once the state becomes true
One event & one or more states	Rule activates when the event occurs, but only if all the states are true
Multiple events	Unlikely that the rule will ever activate
Multiple states	Rule activates as soon as all the states are true

Table 1: The logical meanings we assume trigger combinations have. We expect that a majority of people will have these interpretations of trigger combinations.

wants their lights to turn on when they arrive home, but only after the sun has set. While IFTTT has both a sunset trigger and a personal location trigger, it is not possible to use them both at once to create the desired rule. This limitation has been pointed out in the literature and systems that allow conjunctions of multiple triggers have been developed [19, 4, 8]. However, the distinction between different trigger and action types in the context of multiple triggers has not been studied systematically.

Conjunctions of multiple triggers result in different types of triggers, depending on the types of constituent triggers (Figure 3). Conjunctions of multiple states is another state that is true when all the constituent states are true. Conjunctions of a state and an event is another event that happens at the same time as the constituent event, if the state constituent is true. For example, the rule, “If the doorbell rings and it is between 3:00 - 5:00 pm,” should activate when the when the doorbell rings, but only if it is between 3:00 - 5:00 pm.

A conjunction of two events is another event that happens at the same time as the constituent events, provided that all events occur exactly at the same time, which is theoretically impossible and in practice extremely unlikely. Hence a rule with a trigger like “if the doorbell rings and the sun sets” would almost never activate. Nonetheless, users might actually have an interpretation for such rules. For instance, one could interpret the example above as meaning, “If the doorbell rings *around the time* of the sunset.”

Studying mental model ambiguities

We hypothesize that the lack of distinction between different trigger and action types is a source of ambiguity, especially in the context of trigger conjunctions. In particular, we will point out three potential points of confusion. The first is the interpretation of when exactly triggers will occur. In particular, for state-type triggers, it is unclear if people expect the action to start immediately as the state becomes true, or to happen anytime while the state is true. Second, it is unclear if conjunctions of events (which are practically invalid triggers) are actually meaningful for people. Finally, people’s expectation about whether sustained actions will revert automatically is unknown.

In this work we aim to understand people’s interpretations of these ambiguities when all trigger and action types are supported in the system. To that end we performed two user studies. In the first study, participants were asked to describe their interpretation of given trigger-action programs. This study revealed a significant discrepancy in people’s interpretations (low levels of agreement), where sometimes the selection of the majority did not correspond to the logical interpretation. In the second study, participants were asked to create trigger-action programs for a desired behavior and once again respond to questions related to their interpretation of a given program. This study confirmed that ambiguities are cause for errors; we observed that people created different programs given the same prompt. Furthermore, people’s interpretations of given programs were still in disagreement after having created programs themselves. We describe these two studies in the following sections.

STUDY 1: PROGRAM INTERPRETATION

To begin understanding how users interpret different trigger and action types, we conducted a web-based study on Amazon Mechanical Turk. In the introduction to the study, we introduced the concept of a smart home and gave a few examples of “if-then” rules which could be defined for the home. The study was split into five parts, examining different aspects of TAP.

Interpretation of trigger-action programs consisted of reading a *text description* of the program and then answering multiple-choice questions that assessed the participant’s understanding of the program. These descriptions are akin to text descriptions automatically generated in IFTTT upon the creation of rules (see examples in Fig.1(a)). Event triggers were worded using the active verbs *turns* and *rings*, in line with IFTTT’s way of wording events. State triggers were worded with the present tense of the verb *be*.

Throughout the study, we decided to minimize variance between the questions by using a uniform set of triggers and actions. These are shown in Table 2. The actions were chosen to represent 3 different kinds of actions: sending an email (instantaneous), brewing coffee (extended), and turning the lights on (sustained).

When actions will occur

In the first part of study, we asked users when an action would occur, given single or multiple triggers. The purpose of these

Event triggers	It turns 10:00 am The doorbell rings
State triggers	It is raining It is between 3:00 - 4:00 pm
Actions	Send an email notification Brew a pot of coffee Turn the lights on

Table 2: The set of event triggers, state triggers, and actions used in Study 1.

questions was to evaluate whether users expected actions to occur according to the logical interpretations of the triggers (Table 1). There were 9 such questions: 2 single event triggers, 2 single state triggers, 3 for combinations of one event trigger and one state trigger (we excluded “If it turns 10:00 am and it is between 3:00 - 4:00 pm”), 1 which combined the two event triggers, and 1 which combined the two state triggers. In all cases, the action was simply referred to as “[X],” *e.g.*, “If the doorbell rings, then do [X].”

For each question, the respondents were asked to choose from a multiple choice list to explain when the action would occur. For event triggers, users could say that the action would occur immediately, within 1 minute, or within 10 minutes. For state triggers, users could also say the action would occur at any time while the state was true. For questions with one event and one state trigger, the options for when the action would occur were limited to 1) when the event and state trigger became true simultaneously, or 2) when the state was already true when the event occurred.

For the question with two event triggers, the respondent could say the action occurred if the two events happened at exactly the same time, or within 1 minute of each other, or within 5 minutes of each other. For the question with two state triggers, the respondent could say the action occurred if the first state became true while the second state was already true, or vice versa, or either of those two options. They also could say that the action could occur at any time while the two states were true. For all questions, respondents could also choose to say that the action would not occur at all, or specify a free-response answer.

When actions will end

In the second part of the study, we asked users when an action would end, given a fully specified rule. These questions were designed to study two issues. First, did users believe that sustained actions would automatically end when paired with state triggers (*e.g.*, “If it is between 3:00 - 4:00 pm, turn the lights on”)? Second, did the trigger involved in the rule lead to different expectations about when actions will end?

We gave a fully specified rule for each of the 4 triggers and 3 actions shown in Table 2, for a total of 12 questions. For each question, the respondents could say that the action would end within 1 minute of starting, within 10 minutes of starting, or within an hour of starting. For state triggers, respondents could say that the action would end once the trigger was no longer true. For all questions, the respondents could

say that the action would never end, or they could provide a free-response answer.

Open-ended questions

Next, we asked open-ended questions. Some of the questions were designed to understand people’s interpretation of the differences between different kind of triggers. In particular, we asked respondents to list differences between “If the doorbell rings” and “If it is raining outside,” as well as between “If the doorbell rings” and “If it turns 10:00 am.” We also asked for different wordings of a couple of rules to see if users generally had a preferred way of wording the rule other than an if-then statement.

We asked additional open-ended questions, which were placed at the end of the study, so as not to be leading. These questions directly asked respondents for their opinions on the issues we studied through earlier questions. For example, in one question, we wrote that it would be unlikely for two event triggers to occur at exactly the same time. The users were asked if such rules should be allowed, and what their meaning should be if so. In another question, we asked when an action should occur if the trigger was a state trigger. Finally, we asked, given the rule, “If the time is between 3:00 - 4:00 pm, then turn the lights on,” if the lights should turn off automatically at 4:00 pm, or if there should be another rule to turn the lights off.

Demographic questions

In the last part of the study, we gathered demographic information about the respondents, including their age, gender, level of prior programming experience, and level of prior experience using IFTTT.

The study was formulated as a multi-page questionnaire using Google Forms. Participants were allowed to go back and change their answers.

RESULTS FROM STUDY 1

Demographics

There were 60 respondents to the survey. We restricted the set of workers to those who had obtained the “Master Worker” distinction by Amazon and who lived in the United States. An initial set of 19 respondents were paid \$0.50 to complete the survey. Because the survey took longer to complete than expected, the remaining respondents were paid \$1.00 to complete the survey. 30 respondents were male and 30 respondents were female. The ages of the respondents ranged from 21 to 68 years, with an average of 39.2 and a standard deviation of 11.6. 32 respondents (53.3%) reported no prior programming experience, 19 (31.7%) said they programmed “a little,” and 9 (15.0%) said they programmed “on a regular basis.” 54 (90.0%) respondents said they did not know about IFTTT, 5 (8.3%) said they had heard of it, and 1 (1.7%) respondent said they had used it before.

Findings

Expectations about triggers depend on the specific trigger(s)

We found that respondents had different expectations for when actions should be triggered depending on whether the

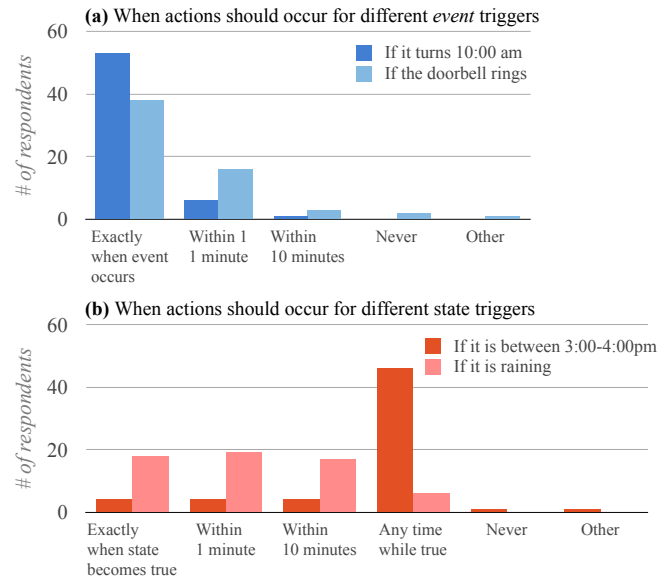


Figure 4: When users expect a rule to activate, for particular (a) event and (b) state triggers. The y-axis indicated the number of participants out of 60.

trigger was an event trigger or a state trigger. Across the two rules that included a single event trigger, 75.8% of responses said the rule would start as soon as the event occurred. But, across the two rules with a single state trigger, only 36.7% of responses said the rule would start as soon as the state became true. This difference can be seen by comparing the leftmost categories in Figures 4(a) and (b).

However, we also found that user expectations varied even between two different event triggers or two different state triggers. Fig. 4(a) shows that when the trigger was “If it turns 10:00 am,” more users thought the rule would start exactly when the event occurred, compared to when the trigger was “if the doorbell rings.” Similarly, for rules with a state trigger, more users thought that the rule would start any time while the state trigger was true if the trigger was “If it is between 3:00 - 4:00 pm,” compared to “If it is raining.”

The free response questions provide the explanation for this apparent inconsistency. Comparing the 10:00 am trigger to the doorbell trigger, one participant wrote “*Doorbell rings are not predictable, whereas times are.*” Similarly, describing rainfall as a trigger, participants wrote, “*When it’s raining, the weather can vary and fade-in/fade-out,*” and “*it’s more ambiguous as to when it can be called rain.*” This could explain why there was more consistency for the triggers involving time.

Participants mostly agreed on the behavior for rules that combined one event trigger and one state trigger. Across the 3 questions of this type, an average of 85% of responses indicated that the rule should activate when the event occurred, as long as the state was true.

Multiple event triggers are considered to be technically valid

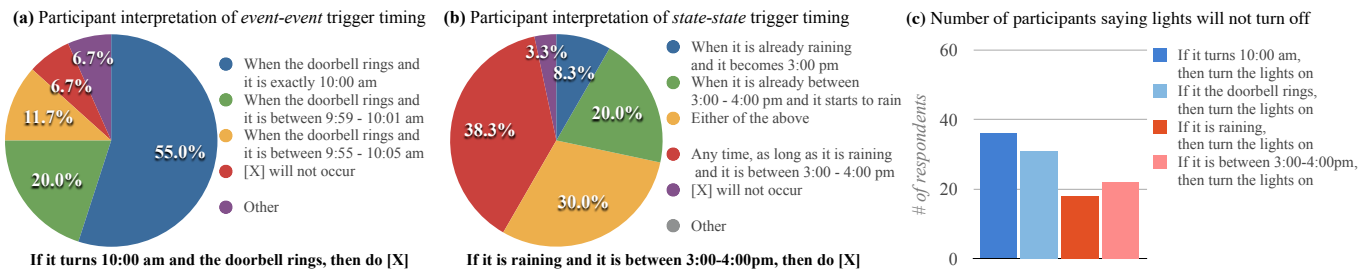


Figure 5: (a) Whether/when users expect a rule with two event triggers to start. (b) Whether/when users expect a rule with two state triggers to start. (c) Whether users expect a sustained action to end, given different event and state triggers.

When asked when the action should start for the rule, “If it turns 10:00 am and the doorbell rings, do [X],” only about 7% of the respondents said that the action would not occur. A majority (55%) said that the rule should only activate if both events occurred simultaneously (Fig. 5(a)). The remaining stated that the rule should activate when one event occurred within 1 or 5 minutes of the other.

In one of the open ended questions, we explained that two event triggers were unlikely to happen at exactly the same time, and asked if such rules should be allowed. Although the wording of the question possibly led them to a particular answer, many respondents gave nuanced responses. For example, one respondent said that the rule should be allowed even if it did not work in practice: “I think they should be allowed, but I can’t really think of many reasons why people would want to do that.” Another participant suggested a way of making it clearer: “I would make the statement require a duration, so within 5 minutes of it starting...” Some participants said that the system could automatically figure out a way to make the rules work, e.g., “Yes, they should be allowed... But their meaning should be as follows: ... So that as long as it is raining as soon as the limited event occurs (doorbell rings), that’s the trigger.”

Expectations varied widely for multiple state triggers

When asked when a rule should start for the trigger, “If it is raining and it is between 3:00 - 4:00 pm,” 38% of respondents said the rule could start *any time*, as long as both states were true, while 30% said the rule would start as soon as both rules *became* true. No single answer had a majority. The results are shown in Figure 5(b).

When sustained actions end depends on the trigger

Across all the rules where the action was “send email” or “brew a pot of coffee,” participants agreed that the action would finish within one or ten minutes, as expected. However, when the action was “turn the lights on,” the responses differed depending on the trigger. When the trigger was an event trigger, an average of 56% of respondents said the light would not turn off. However, when the trigger was a state trigger, an average of only 33% of respondents said the light would turn off. The data is shown in Figure 5(c).

STUDY 2: PROGRAM CREATION

Our first study showed that users’ interpretations of trigger-action programs for different trigger combinations and sustained actions differed from the logical interpretation we expected. However, this study did not ask users to synthesize rules themselves. While interpretation of programs from descriptions is relevant for many IFTTT users who browse, select, and activate shared programs without modifying them, many other users create programs themselves. Seeing the rule creation process in the context of a fully implemented interface could positively impact the user’s mental model of how the program should behave. To investigate whether program creation mitigates the ambiguities observed in the first study, we designed a TAP interface and conducted a second study.

Interface design

Our interface was designed to feature multiple triggers with different trigger and action types, while also resembling IFTTT. The interface borrows visual aspects of IFTTT, as well as the workflow for creating rules. This allows the results and recommendations from studying this interface to be applicable to a general set of similar interfaces.

Choice of triggers and actions

The interface supported a set of 5 trigger and 5 action categories. The triggers and actions were generic smart home capabilities and did not specify any real-world products. Some trigger categories supported both event triggers and state triggers. For example, the “My location” trigger category could be made into either “I arrive at work” or “I am currently at work.” Other trigger categories only supported event triggers, such as the “Doorbell rings” trigger. Similarly, some of the action categories could be made into only sustained actions, only non-sustained actions, or both. A full list of triggers and actions is listed in Table 3.

Multiple triggers

Because IFTTT does not support multiple triggers, incorporating this aspect into our interface was the most open-ended design work we engaged in. One observation about the IFTTT interface is that throughout the rule creation process, it shows partially completed previews of the rule, with a single clickable link to proceed. We decided to replicate this design by offering users a choice after the first trigger was added: add another trigger by clicking “and this” or select an action by click “then that.” This process repeated for as many

Triggers	Actions
Daily time (event, state)	Switch lights (sustained)
Weather (event, state)	Brew coffee (non-sustained)
Doorbell rings (event)	Doorbell rings (sustained)
My location (event, state)	My location (non-sustained)
Motion detector (event)	Motion detector (sustained, non-sustained)

Table 3: Trigger and action categories in our TAP interface.



Figure 6: A composite screenshot of the rule creation flow in (a) IFTTT and (b) our interface.

triggers the user wanted to add. Fig. 6(a) and (b) show how this process looks in IFTTT and in our interface.

The interface design went through several iterations, starting with paper prototyping. We simulated the interface on paper with several participants and informally gathered feedback on the design. During the paper prototyping, we asked participants to create 3 rules, some of which required multiple triggers. No participant had major difficulties with the paper prototypes, although minor adjustments were made between prototypes based on feedback. In the next phase, we implemented a digital prototype of the interface and gathered further informal feedback. Finally, the interface was incorporated into a user study interface.

Wording of triggers and actions

Unlike IFTTT, which offers both an icon and a name for each trigger and action, our interface only included a textual name that describes the trigger and action. These were chosen to be self evident and to clearly communicate the trigger and action type. As in our first study, event triggers included active verbs (e.g., “I arrive at,” “I leave”) as in IFTTT, while state triggers included the verb *be* (e.g., “I am currently at”). As in IFTTT, our interface involved first choosing a trigger category and then choosing the particular trigger from a drop-down menu.

Questionnaire

The second study contained 5 program creation questions (P1-5, Table 4) and 5 multiple choice questions about the participant’s interpretation of a given rule (Q1-5, Table 5). For program creation tasks, participants were given a description of a desired behavior and were asked to create one or more rules for the smart home to achieve that behavior. Each of these questions was worded to avoid using the words “if” and “then.” We also avoided phrasing desired behaviors in ways that could have direct mappings to the interface. Program interpretation tasks were similar to Study 1; users were given

Trigger types	Program behavior description
Single event trigger	P1: You want the lights to turn on at 6:00 pm every day
One event and one state trigger	P2: You want to be notified, via email, should a person be detected in the house while everyone’s at work (9:00 am - 5:00 pm every day)
A rule that could involve two event triggers	P3: Your work starts at 9:00 am. On days when you get to work on time, you want to send an email to yourself saying “I got to work on time!”
Single state trigger	P4: You want the thermostat to be off as much as possible, unless the temperature outside is below 40 degrees, in which case the thermostat should be set to 72 degrees.
Two state triggers	P5: You want a pot of coffee to be brewed when it’s below 40 degrees outside, but only before 10:00 am every day.

Table 4: Program behaviors that users were asked to create rules for in Study 2.

Rule	Multiple choice question
Q1: If the time is 6:00 pm, then turn the lights on	If this is the only rule, do you expect the lights to turn off, and if so, when?
Q2: If I arrive at home and the time is between 6:00 - 11:00 pm, then turn the lights on	If this is the only rule, and you arrive home at 5:00 pm, do you expect the lights to turn on, and if so, when?
Q3: If the doorbell rings and the time is 3:00 pm, then unlock the front door for 10 seconds	You are expecting visitors to your house at 3:00 pm. If you wanted them to be let in automatically, would you use this rule? If so, when do you expect the door will unlock?
Q4: If it is snowing, then turn the thermostat to 75 degrees F	When will the thermostat be set to 75 degrees and when will it turn off?
Q5: If the time is between 7:00 am and 10:00 am and the outside temperature is below 40 degrees, then brew a pot of coffee	Suppose it was cold all night. Do you expect the coffee brewer to start, and if so, when?

Table 5: Multiple choice questions in Study 2.

textual descriptions of rules programmed in the interface, and were asked a multiple-choice question about the rule.

The program behaviors and the interpretation questions were designed to cover a variety a trigger combinations (Table 4, left column). Some questions involved a sustained action, while others did not. For both sets of 5 questions, a sustained action was paired at least once with an event trigger, and with a state trigger.

The 5 program interpretation questions were asked after the 5 program creation tasks. This is because the questions could call attention to issues such as whether it makes sense to have two event triggers in the same rule, which could affect the rules that the participant would create later on. Users were not allowed to change their answers to previous questions in the study. Finally, we collected the same basic demographic information as in the first study.

RESULTS FROM STUDY 2

Deployment

The user study was distributed through Mechanical Turk. Workers were limited to “Master Workers” who lived in the United States. Each participant was paid \$1.50 to complete the survey, which took about 20 minutes to complete. We advertised the study as being about “programming rules for a smart home.” Prior to beginning the study, participants were given a short introduction to the concept of smart homes, and given some examples of their capabilities. They were also asked to assume that the smart home was capable of reliably executing all the sensing and actions shown in the interface.

Demographics

There were 42 participants who completed the study. The ages of the respondents ranged from 20 to 66 years, with an average of 37.45 and a standard deviation of 10.9. 22 respondents (52%) were male, and 20 (48%) were female. 22 (52%) respondents said they had no programming experience, 17 (40%) said they had “a little” programming experience, and 3 (7%) said they had programming experience. 34 (81%) said they had not heard of IFTTT before, 8 (19%) said they had heard of it, and none said they had used it before.

Findings

Multiple event triggers were used in practice

In the creation of P3 (Table 4), 21 (50%) respondents created the rule, “If I arrive at work and it turns 9:00 am, then send myself an email,” as opposed to “If I am currently at work and it turns 9:00 am”, which only 9 (21%) respondents made (Fig. 7(a)). We consider the first rule to be incorrect, both because it uses two event triggers and it would not fire if the person was on time by being at work early.

One reason why this could have happened is that users do not naturally think of “arriving at work” to be an event at a specific point in time, but as a state which is true all day as soon as you arrive at work. In our interface, the “am currently at” and “arrive at” options were placed adjacent to each other, so users are likely to have seen both options. However, this raises a thematic issue with interfaces like IFTTT, which provide a natural language mapping from the interface to program behavior. Because natural language itself can often be ambiguous, the meaning of the programs they describe can become unclear as well.

Q3 (Table 5) also demonstrates that people are okay with multiple event triggers. 15 (36%) respondents said that they would use the rule shown, and that the door would open if the doorbell rang at exactly 3:00 pm. 13 (31%) respondents said they would use this rule, and that the door would open if the doorbell rang between 3:00 - 3:01 pm. Only 13 (31%) respondents said that they would not use the given rule. This shows that most users believed that the rule would work, either because they expected the visitors would arrive very promptly at 3:00 pm, or because the system would work even if the visitors were early or late by a few minutes.

Event & state triggers were hard to reason about

We were surprised to find that the composition of an event trigger with a state trigger was not well understood. For P3

(Table 4), the ideal combination of triggers would have either been “If it turns 9:00 am and I am currently at work,” or alternatively “If the time is between 8:00 am and 9:00 am and I arrive at work,” both of which are combinations of an event trigger and a state trigger. However, only 9 (21%) users created the first rule, and 3 (7%) created a rule similar to the second rule (shown as “Other correct” in Figure 7(a)). This could suggest that rules with a combination of an event and a state trigger are hard for users to synthesize.

Additionally, in response to Q2 (Table 5), only 24 (57%) participants said that the lights would not turn on, while the remainder said that the lights would turn on, either any time between 6:00 - 11:00 pm or exactly at 6:00 pm (Figure 7(c)).

Users had varied mental models for state triggers

Of the two minority responses to Q2, 10 (24%) participants said that the lights would turn on any time between 6:00 - 11:00 pm, and 8 (19%) said that the lights would turn on at 6:00 pm. This shows that users are still not sure if state triggers activate as soon as the state becomes true, or at any time while the state is true.

Similarly, for Q4, 33 (79%) users said the thermostat would be set as soon as it started snowing, while the remaining 9 (21%) said that the thermostat would be set any time while it was snowing (Figure 7(d)). Although a majority answered in the way we expected, in a real-world deployment, having 21% of users misunderstand this type of trigger would be a non-trivial problem.

In response to Q5, 26 (62%) participants said that the coffee brewer would start at 7:00 am, while 15 (36%) said that the coffee brewer would start any time between 7:00 - 10:00 am. These results are consistent with Study 1, showing that most users expect state triggers to activate as soon as all the states became true, but a non-trivial percentage of users had a different interpretation.

Users disagreed on sustained actions and forgot to undo them

While programming P4 (Table 4), users were asked to keep the thermostat off as much as possible, unless the temperature outside dropped below a certain level. However, most users forgot to turn the thermostat off once they had set it. 33 (79%) participants made the rule, “If the temperature outside is below 40 degrees, then set the thermostat to 72 degrees,” but only 6 participants made both that rule and another rule, “If the temperature outside is above 40 degrees, then turn the thermostat off.”

The multiple-choice questions revealed that more people thought that sustained actions would be undone automatically when the trigger was a state, compared to when the trigger was an event (Figure 7 (b,d)). Q1 and Q4 asked users when a sustained action would end. For Q1 (event trigger) 39 (93%) respondents said that the lights would not turn off. However, for Q4 (state trigger) only 27 (64%) respondents said that the thermostat would not turn off. Q4 also shows that users did not universally agree whether the thermostat should turn off by itself or not, as 15 respondents (36%) said the thermostat would turn off as soon as it stopped snowing.

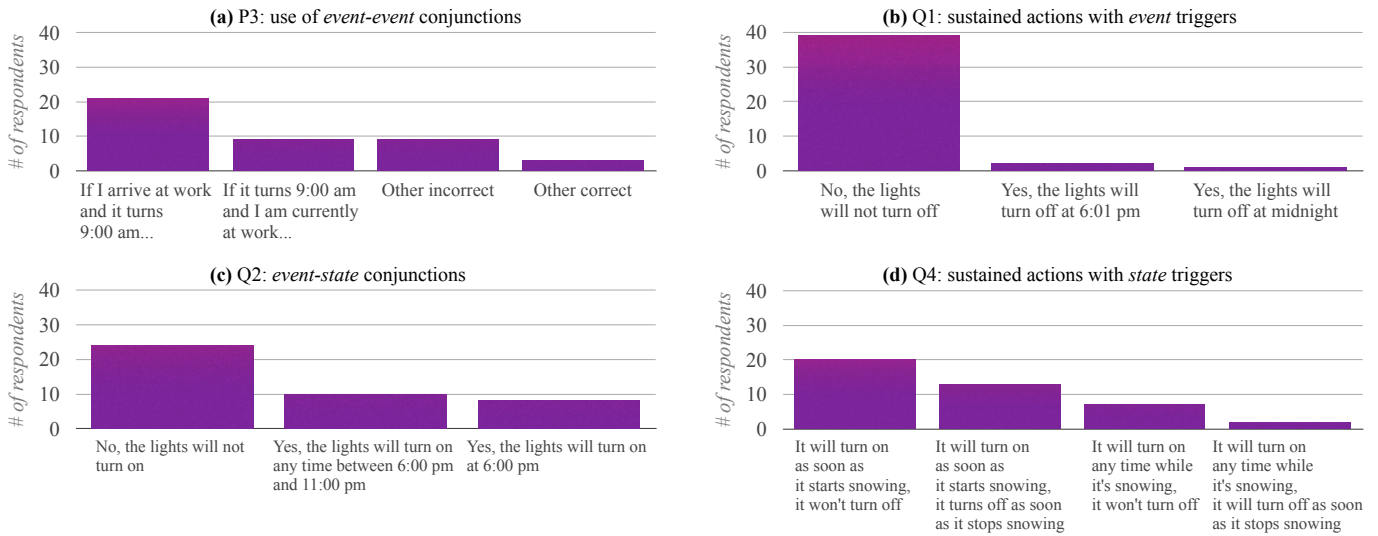


Figure 7: (a) The distribution of rules that users created for P3 and (b-d) distribution of responses users gave to Q1, Q2, and Q4 in Study 2. The y-axis indicates number of participants out of 42 on all graphs.

User interpretations may be influenced by existing products
 In the creation of P2, only 30 (71%) respondents created what we considered to be the correct rule, “If it is between 9:00 am - 5:00 pm and motion is detected, then send myself an email.” The most common mistake was to not limit the time the rule was active, as in, “If motion is detected, then send myself an email,” This rule was made by 8 (19%) participants. We speculate that some of those participants did not feel the need to specify a time range condition themselves, since many home security systems are manually deactivated through a passcode when an occupant arrives home.

DISCUSSION

Our studies were designed to assess the accuracy of people’s mental models of trigger-action programs in the presence of different trigger and action types within a system that allows conjunctions of multiple triggers. Our emphasis, in comparison to previous user studies involving TAP, is on the distinction of different trigger and action types. While the differences between these types impact the underlying meaning of programs, previous work or existing systems do not make a distinction at the interface level. Our studies reveal that this causes ambiguities in interpreting meanings of programs and errors in creating programs with an intended behavior.

Problems due to mental model inaccuracies

Based on our two studies we make the following high-level characterization of mental-model problems:

- **Trigger timing:** Participants disagreed on whether state triggers start as soon as the state becomes true, or any time while the state is true. Furthermore, users give fuzzier triggers like rainfall or the doorbell ringing leeway in terms of start time.
- **Program validity:** Participants disagreed on whether multiple events should be allowed.

- **Action reversal:** Participants disagreed on whether sustained actions end automatically when paired with a state trigger. They did not add rules to undo sustained actions, likely as a consequence of believing that the actions would end automatically.

We observed similar problems in both of our studies, suggesting that program creation does not improve the users’ mental model so as to mitigate issues that exist in program interpretation.

Interface improvements

Next, we describe four different interface adaptations that could address some of these problems.

Prompts: One way to mitigate some of the common issues would be to include prompts to warn users in situations that were empirically demonstrated to cause ambiguities. These prompts would inform the user about the true semantics of the programs they create. For example the system could: (i) warn users that two events are unlikely to happen exactly at the same time, after the user adds a second event trigger; (ii) tell users when the action will start (e.g., immediately or within a minute) when they add a single state trigger or “fuzzy” event trigger; (iii) warn users that an action will or will not automatically revert when they add a sustained action. Such prompts could also offer semi-automated ways for addressing the ambiguity. For instance, when a rule with a sustained action is created, the prompt could allow easy creation of a second rule that undoes the action.

Disallowing confusing options: Another mechanism would be to disallow the creation of programs that are invalid with respect to the true semantics of the particular TAP system. For instance, our system could disallow creation of programs that have (i) more than one event trigger, or (ii) purely state-type triggers.

```

IF event-trigger THEN action
WHEN event-trigger DO action
WHILE state-trigger DO sustained-action
AS LONG AS state-trigger DO sustained-action
IF event-trigger WHILE state-trigger THEN action
IF state-trigger WHEN event-trigger THEN action
WHILE state-trigger AND state-trigger DO sustained-action
AS LONG AS state-trigger AND state-trigger DO sustained-
action OTHERWISE DO  $\neg$ sustained-action

```

Table 6: Alternative high-level program statements for particular trigger and action type combinations.

Trigger duality: Conjunctions of triggers are meaningful for combining a single event trigger and multiple state triggers. To support combinations of all categories of triggers it is important for the system to provide both state and event triggers related to the same underlying concepts; e.g., include all three of “it starts raining,” “it is raining,” and “it stops raining.” For triggers that are naturally expressed as states, the system can provide all state changes as events. For triggers that are naturally expressed as events, states could be defined by adding an adjustable time window around the event. For example, “the doorbell rings” event could become “the doorbell rang in the past 2 minutes” state. While the duality of states and events could naturally draw users’ attention to the distinction between the two types, making the categorical distinction at the interface level (through grouping or actually naming trigger types as states and events) could further improve the user’s mental model.

Top-level statements: One of the key ways in which TAP interfaces achieve simplicity is by mapping natural language elements (e.g., “if” and “then”) to program behavior. This mapping can create ambiguity, since natural language itself is inherently ambiguous. At the same time, humans are good at refining natural language statements to more accurately communicate an intended meaning. Similarly, we propose extending TAP interfaces to support alternative high-level statements that more accurately indicate the type of triggers and actions they support. Some examples for two-trigger statements are given in Table 6. These specific statements could make it easier for people to combine the right types of triggers or remember to undo the effect of sustained actions.

The mechanisms proposed above individually address a subset of the problems observed in our study. Therefore, mitigating all of issues will require combining these mechanism. Determining the right combination, as well as the details of each mechanism, such as which high-level statements should be included, require further design and empirical evaluation. We hope to tackle these questions in our future work.

Limitations

Our work has several limitations. In both of our studies, we asked participants to create or answer questions about programs that were not their own idea, so the intent of the behaviors we described may not have been clear. The interface we tested was designed to resemble IFTTT. This limits our

findings and results to interfaces which are similar to our interface or to IFTTT. Also, our subjects were anonymous users from Mechanical Turk, who may not be representative of potential users of TAP systems.

As we saw throughout the study, wording played an important role in how users interpreted rules. This suggests that our results could be dependent on the exact choice of wording we used in our questions and in our interface. For example, in question 2 of the user study, we say “while everyone’s at work (9:00 am - 5:00 pm every day),” which could be interpreted as either a “My location” trigger or a “Daily time” trigger, or both. Similarly, we discussed in our user study findings how the meaning of “If I arrive at” versus “If I am currently at” could have differed between respondents.

CONCLUSION

This work aims to understand people’s interpretations of ambiguities in trigger-action programming (TAP) systems, due to the lack of a distinction between different trigger and action types. We performed two user studies to verify ambiguities and characterize their consequences. In the first study, participants were asked to describe their interpretation of given trigger-action programs. This study revealed a significant discrepancy in people’s interpretations, often deviating from the actual semantics of the program. In the second study, participants were asked to create trigger-action programs for a desired behavior and once again responded to questions about their interpretation of a given program. This study confirmed that ambiguities are cause for errors, demonstrating that people create different programs given the same prompt and are still in disagreement in their interpretations after having created programs themselves. Finally, based on our results, we presented potential interface adaptations for improving TAP interfaces, so as to mitigate inaccuracies in the mental models of users.

REFERENCES

1. Alexandrova, S., Tatlock, Z., and Cakmak, M. RoboFlow: A flow-based visual programming language for mobile manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)* (2015).
2. Biggs, G., and MacDonald, B. A survey of robot programming systems. In *Proceedings of the Australasian Conference on Robotics and Automation* (2003), 1–3.
3. Dabek, F., Zeldovich, N., Kaashoek, F., Mazières, D., and Morris, R. Event-driven programming for robust software. In *Proceedings of the 10th Workshop on ACM SIGOPS European Workshop*, ACM (2002), 186–189.
4. Dey, A. K., Sohn, T., Streng, S., and Kodama, J. iCAP: Interactive prototyping of context-aware applications. In *Pervasive Computing*. Springer, 2006, 254–271.
5. Edwards, W. K., and Grinter, R. E. At home with ubiquitous computing: Seven challenges. In *UbiComp 2001: Ubiquitous Computing*, Springer (2001), 256–272.

6. Fischer, G., and Giaccardi, E. Meta-design: A framework for the future of end-user development. In *End User Development*. Springer, 2006, 427–457.
7. Gross, P., and Powers, K. Evaluating assessments of novice programming environments. In *Proceedings of the First International Workshop on Computing Education Research*, ACM (2005), 99–110.
8. Häkkinä, J., Korpipää, P., Ronkainen, S., and Tuomela, U. Interaction and end-user programming with a context-aware mobile application. In *Human-Computer Interaction-INTERACT 2005*. Springer, 2005, 927–937.
9. Hanson, E. N., and Widom, J. An overview of production rules in database systems. *The Knowledge Engineering Review* 8, 02 (1993), 121–143.
10. Jahnke, J. H., d'Entremont, M., and Stier, J. Facilitating the programming of the smart home. *Wireless Communications, IEEE* 9, 6 (2002), 70–76.
11. Karat, C.-M., Karat, J., Brodie, C., and Feng, J. Evaluating interfaces for privacy policy rule authoring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2006), 83–92.
12. Kim, S., and Jeon, J. Programming LEGO Mindstorms NXT with visual programming. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, IEEE (2007), 2468–2472.
13. Ko, A. J., Myers, B. A., and Aung, H. H. Six learning barriers in end-user programming systems. In *Visual Languages and Human Centric Computing, 2004 IEEE Symposium on*, IEEE (2004), 199–206.
14. Lin, J., Wong, J., Nichols, J., Cypher, A., and Lau, T. A. End-user programming of mashups with Vegemite. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, ACM (2009), 97–106.
15. Mackay, W. E., Malone, T. W., Crowston, K., Rao, R., Rosenblitt, D., and Card, S. K. *How do experienced Information Lens users use rules?*, vol. 20. ACM, 1989.
16. Myers, B. A., Ko, A. J., and Burnett, M. M. Invited research overview: end-user programming. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, ACM (2006), 75–80.
17. Pane, J. F., Myers, B. A., and Miller, L. B. Using HCI techniques to design a more usable programming system. In *Human Centric Computing Languages and Environments, 2002. Proceedings. IEEE 2002 Symposium on*, IEEE (2002), 198–206.
18. Truong, K. N., Huang, E. M., and Abowd, G. D. CAMP: A magnetic poetry interface for end-user programming of capture applications for the home. In *UbiComp 2004: Ubiquitous Computing*. Springer, 2004, 143–160.
19. Ur, B., McManus, E., Pak Yong Ho, M., and Littman, M. L. Practical trigger-action programming in the smart home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2014), 803–812.
20. Welbourne, E., Balazinska, M., Borriello, G., and Fogarty, J. Specification and verification of complex location events with panoramic. In *Pervasive Computing*. Springer, 2010, 57–75.
21. Zhang, H., and Boyles, M. J. Visual exploration and analysis of human-robot interaction rules. In *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics (2013), 86540E–86540E.