Locked or Not? Mental Models of IoT Feature Interaction

Svetlana Yarosh
University of Minnesota
Computer Science & Engineering
lana@umn.edu

Pamela Zave
AT&T Labs Research
Bedminster, NJ, USA
pamela@research.att.com

ABSTRACT

Internet of Things (IoT) frequently involves conflicting interactions between devices and features that must be resolved to a single system state. The problem of feature interaction (FI) resolution has been investigated in Software Engineering through approaches that focus on verifiability but usually do not include the user in the evaluation. This paper bridges the gap between IoT approaches in HCI and Software Engineering by applying qualitative methods to understanding users' mental models of one representative FI resolution mechanism. Our contributions are in identifying common mental model errors and biases and how these may inform future IoT systems and research.

Author Keywords

IoT; home automation; feature interaction; mental models

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces: *User-Centered Design*.

INTRODUCTION

Edwards and Grinter's groundbreaking work on Ubiquitous Computing in the home articulated "impromptu interoperability" of devices in a smart home as a key challenge for home technologies [5]. Researchers in Software Engineering (SE) have been investigating a similar challenge in many other domains (e.g., telecommunications [20]) and typically address it through a feature-composition approach. To understand this issue, consider the following scenario:

> *Jill's new smart home door lock seemed like a good idea when all it did was provide convenient keypad access. But then, her husband installed a new feature to keep it automatically locked at night. Next, her daughter installed a new driveway sensor to unlock the door if the family car drove up. Now, as Jill is driving away from her house in the evening, she is confused. 'Is my door locked or unlocked?' she worries.*

The behavior of Jill's door lock and many other complex IoT systems can be described in terms of largely independent increments called *features*. A feature may have its own purpose, triggering situation, or intended beneficiary. In many IoT applications (as in Jill's scenario), multiple rules may attempt to influence the state of one actuator. A *feature interaction* (FI) is a logical conflict in determining the value of a particular actuator or system output. If FIs are not resolved before system implementation in a comprehensive analysis phase, then they must be resolved at runtime to produce well-defined system behavior. As long as FIs are resolved in a verifiable way, new features and devices can be added to such a system through integration with the central controller but without the need to modify or inform any existing features or devices of the upgrades.

Software Engineers have investigated FI resolution as a major challenge in creating secure and verifiable IoT systems. However, most SE investigations do not validate their proposed solutions with users. On the other hand, HCI takes a user-centered approach, but has not addressed issues of FI resolution. In this note, we seek to bridge the two communities. We apply HCI methods to understand user mental models of a FI resolution approach (one developed in industry and integrated into commercial home automation systems). Our investigation focuses on two questions:

R1: How well do participants understand the example FI resolution mechanism after an initial exposure?
R2: What common mental model errors do participants make when beginning to understand FI resolution?

In this note, we describe a mixed-methods lab-based investigation where we elicited user mental models of the resolution mechanism by asking users to respond to twenty IoT home automation scenarios. Our results focus on the mental model errors and biases common in the way users perceive FI resolution. In the discussion, we reflect on lessons for both HCI and SE researchers as we collectively advance IoT in the home.

RELATED WORK

Three bodies of related work inspired this study: home automation in HCI, qualitative methods for understanding users' mental models, and feature interaction in SE. Our first major influence is the significant body of HCI investigations on home automation. Some previous work in this domain has examined user perceptions and practices around existing systems, such as the Nest thermostat [19]. Others look more holistically at the role of home automation, such as how power users find satisfaction in their ability to automate various aspects of their daily life [17]. Most of these studies confirm that home automation is notoriously chal-

CHI 2017, May 06-11, 2017, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/3025453.3025617

lenging for the user to set up, control, and secure [2]. In addition to studying existing systems, HCI researchers have designed new approaches to home automation. Most of these focus on novel interaction with a single home automation feature (e.g., [10]), on supporting user scripting of system events (e.g., [9], [13]), and on developing infrastructures of sensors and actuators (e.g., [1]). All of these systems are likely to face FI resolution issues [15], which have largely been left as an exercise to the user to resolve.

Second, we were inspired by methodological approaches for understanding users' mental models of complex systems. Rasmussen's work highlights that the user must represent the internal structure of a system with a "mental model" which may help drive goal-oriented action [12]. Our research questions in this work focus on eliciting the users' initial mental models of FI resolution mechanism infrastructure by understanding the errors they make in explaining the mechanism's functionality. This study was inspired by work on folk understanding of home control interfaces, namely Kempton's use of qualitative interview methods to identify two distinct mental models of home heating control [7]. Like Kempton, we turn to qualitative methods to reveal some of the assumptions inherent in the SE approach, by highlighting situations where the user's mental model is in conflict with the actual behavior of the system.

Finally, feature interaction (FI) is a subfield of SE, traditionally focused on addressing the multi-device and multi-feature needs of telecommunication technology (e.g., [20]). Most SE approaches have focused on detecting undesirable FI [8,16], assertion-based testing of behavior during FI [11], and runtime verifiability [18]. However, while these approaches present a number of sound and verifiable solutions, none of them have been validated in with user-centered methods.

RESOLUTION MECHANISM FOR FI

We chose one representative FI resolution mechanism to use in our study. The description of the resolution mechanism and features came from our previous work in SE, which established this as a verifiable and scalable approach to FI resolution [21,22]. It is the approach currently considered for real-world IoT systems, like AT&T's Digital Life [23]. We applied this resolution mechanism to a hypothetical smart lock with four features (in order of priority): (1) a lock/unlock function keypad (2) an intruder defense function that will attempt to lock the doors if sensors detect an intruder, (3) a hands-free operation function that will attempt to unlock the door if it senses the resident's car in the driveway, and (4) a night lock that will attempt to lock the door during hours specified as night.

These features were described to users in a manual (see supplementary materials), along with the three key aspects of the FI resolution mechanism: priority, duration, and "don't care means no change:"

Priority: in some situations one feature might want the door locked for security, while another feature wants the door unlocked for convenience. We manage these situations by allowing features to override lower-priority features. At any time, the actuator setting is determined by the preferences of the highest-priority feature.

Duration: To make the priorities work correctly, we need the concept that each operation has "duration." When the duration of an operation is over, the feature no longer affects whether the door is open or not.

"Don't care means no change:" After an operation's duration expires, the rule is "don't care means no change." For example, if the unlocking feature does not care anymore because the duration period is over, and if no other feature wants the door locked, then the door will stay unlocked until some feature does want the door locked.

METHODS

We conducted a mixed methods lab investigation of participant mental models of the FI resolution mechanism described above. Twenty adult participants (12 female, 8 male) with diverse technical skills and occupations (see supplementary materials) participated. We deemed that data saturation was reached before getting to 20 participants, as by participant 16 we began seeing the mental model errors repeating and no new ones introduced. After each participant expressed that he or she had a clear understanding of the resolution mechanism, we posed 20 scenarios (see supplementary materials), asking him or her to specify whether the door is locked or unlocked in the described situation and explain why they think so.

One example of a scenario is below:

> *According to your house schedule, it is night. You drive home and park and then walk quickly to the door. Is the door locked or unlocked? [Correct Answer: Unlocked]*

Participants were not told the correct response during the scenario part of the study, but could ask how they did during the debriefing.

Qualitative data-driven analysis was carried out using Seidman's thematic analysis guidelines [14]. All study sessions were transcribed and open-coded by the lead author. The lead author and another research clustered codes through affinity mapping. Once the two researchers achieved consensus on the error codebook, the lead author coded all the transcripts using this guide. We triangulated quantitative data (e.g., questions gotten right) with the qualitative data from participant responses to understand the participants' mental models of the FI resolution mechanism.

RESULTS

R1: FI Resolution Mechanism Understanding

On average, participants got 88% of the questions right (17.5 questions, *SD*=1.4). This may seem fairly high, but needs to be considered in the context of two factors: (1) since the responses were binary, random guessing would have produced a 50% success rate, and (2) every participant got at least one question wrong. Clearly, understanding FI resolution is a fairly complex task.

Many participants made errors that could be classified as "careless:" misreading the manual or scenario, forgetting about a feature of the system, or confusing the order of the priorities. However, a greater number of participants made mistakes that pointed to persistent errors in their mental models of the FI resolution mechanism. Table 1 reports the frequency of all of these mistakes.

Many participants had trouble understanding or explaining the system, e.g.:

> *I'm not sure how I would be able to explain it ... Sometimes my door is locked or unlocked. I'm not sure when.* -P6

This is not how any person wants to feel about the front door of his or her house. There were also 31 separate cases (out of 400) in the study where a participant predicted that the door would be locked but it was actually unlocked based on the model. These are potential security hazards. It is clear from this investigation that a short written description of the basic interaction model and its features does not provide adequate exposure for all participants to be confident and correct in their understanding.

R2: Common Mental Model Errors & Biases

To understand the mental models of our participants, we holistically examined their responses to scenarios and the verbal explanations they gave for their answers. Two scenarios emerged as being particularly important to developing this understanding and we refer to them throughout:

> ***Grocery Scenario.*** *[During the day.] You drive home and park. Your car is full of groceries and other shopping, which take many trips to bring into the house. Five minutes after you drove in, you are still making trips to the car. Is the door locked or unlocked? [Correct Answer: Unlocked]*
>
> ***Weekend Scenario.*** *You have set the night period (for Night Lock) to begin at 11 p.m. and end at 7 a.m. On Saturday morning, you get up at 9 a.m. At that time, is the door locked or unlocked? [Correct Answer: Locked]*

On the surface, these two scenarios test very similar ideas: they both ask about a single feature and both require an understanding of the "don't care means no change" aspect of the resolution mechanism. However, only 3 out of the 20 participants answered the grocery scenario correctly. The weekend scenario was the second most incorrectly answered question, but 15 out of the 20 participants were able to get it correct. With in-depth analysis, we came to understand the mental model errors that led to this result.

Feature as Toggle

Some participants seemed to have trouble understanding the idea of "don't care means no change." Instead, these 20% of participants thought that after the duration for an operation expires, the system would toggle the state of the door. This mistake was most commonly articulated in talking about hands-free operation, leading to a wrong answer to the grocery scenario question, e.g.:

> *The door would be locked after that point because three minutes have gone by. It unlocks for three minutes and then locks afterwards.* -P9

Because these participants did not understand the "don't care means no change" mechanism, we would also expect them to get the weekend morning scenario wrong. That is in fact what we saw in the data—all of these participants expected the door to toggle to "unlocked" in the weekend morning scenario.

Sensor Events as Interrupts

Another common mistake was in considering sensor events (intruder alert, car driving up) as different from user-initiated or user-scheduled events (pushing the lock button, scheduling a night lock). Though the 35% of participants who made this error generally understood the idea of "don't care means no change," they thought that "no change" after a sensor event duration expired meant reverting to the previous state, e.g.:

> *I guess the answer depends on whether the door was locked to begin with. If it was locked right before the intruder was detected, it has no reason to automatically unlock the door, but if the door* ***was*** *unlocked then several minutes later, it should have returned it to the state where it was beforehand.* -P18

This was another common reason for a wrong answer to the grocery scenario since the participant thought of the door as being locked before the hands-free sensor event interrupted this state, e.g.:

> *I think based on the manual that it would lock after those 3 minutes because it was locked before.* -P12

Since the night lock feature was perceived as a user-scheduled event, we would expect these participants to give the correct answer to the weekend scenario. All of them did.

Invisible Default

Though the word "default" was never used in the manual or suggested by the researcher, 25% of the participants assumed that there must be a system default and answered the questions with this default in mind, e.g.:

> *It looks like it defaults to locked unless you do something else and it's only unlocked for a minute or two.* -P5

Participants who made the default error generally assumed a locked default and thus gave the wrong response to the grocery scenario. All of these participants gave the correct

	Type of Error Made	% People
General Task Complexity Issues	Careless error reading	10%
	Forgets a feature	25%
	Confuses priorities	25%
Mental Model Mistakes	Feature as toggle	20%
	Sensor events as interrupts	35%
	Invisible default	25%
	Articulated lock bias	50%

Table 1. User error types

answer to the weekend scenario, but the wrong reason, e.g.:

> *I think it's still locked, because the program ran out but the default is still locked.* -P15

They were not able to see that the door was locked not because of a lock default but simply because the night lock feature was no longer expressing a preference and no other feature had issued an unlock request.

Articulated Lock Bias

The most common source of errors made by participants was a bias against suggesting that the door state would be unlocked. Most frequently, it was expressed by making statements such as "*it should be locked because there's been nothing to indicate an unlock*" (P4). One participant got all questions right except the grocery scenario. She was surprised to learn that she got this question wrong and explicitly articulated this lock bias:

> *I guess it makes sense to me when the door would stay* ***locked*** *until something changes, but it seems weirder that the door would stay* ***unlocked****...* -P17

Unlike other mental model errors, where participants made the same mistake consistently, the lock bias merely stated that they were less sure of the answer when it was unlocked. Participants who expressed this bias correctly answered 94% of "locked" scenarios, they were only successful in 72% of the "unlocked" ones. They were more conservative with saying that the door was unlocked than saying that it was locked.

DISCUSSION

Mental Models and Biases in Specific Contexts

While HCI has not considered FI (beyond basic if-then triggers [6]), SE typically considers feature interaction resolution mechanisms independent of the domain and features involved (e.g., [3]). Feature interaction resolution mechanisms that work for telecommunications should apply to home automations or to enterprise warehouse management. However, while that is true for questions of verifiability, our study shows that the context and features involved *may* impact the users' understanding of the behavior of the resolution mechanism. We found three context-dependent biases to be particularly salient.

First, system states may have meanings. To users in this study, the binary states of the system—locked or unlocked—were not functionally equal or equally likely. The idea of an "unlocked door" carried inherent negative meaning for at least half of the participants in the study. Users may carry similar biases in approaching system states in other domains and these must be understood and addressed in order to create predictable systems that users can trust.

Second, user- and sensor-triggered events may carry different expectations. For example, more than a third of our participants expected the system to return to the last user-specified state after the duration of a sensor event expired, even though they otherwise understood the "don't care means no change" aspect of the resolution mechanism.

Third, powerful concepts like "system default" may be embedded in the particular domain. The idea of default was so strongly embedded in the participants' understanding of how IoT works, that some users were *sure* they had seen it *somewhere* in the manual. Just as an optical illusion may make people see motion or depth where there is none, a bias like the expectation of a default made participants remember seeing it even though it was not in the manual. "Default" is just one powerful example and one that was most salient in this study and context, however other feature interaction contexts may come with their own embedded biases. **Based on our findings, we recommend that researchers conduct evaluations of potential FI resolution mechanisms *in specific contexts of use* in order to respond to users' mental models in each domain.** This could help anticipate and account for specific errors due to inherent meanings of states, emphases on some types of triggers over others, and embedded biases like "default."

Lessons for HCI

IoT device scripting and home automation are notoriously challenging for most users [17], but FI resolution mechanisms may be able to provide a good metaphor for complex systems. While the HCI community has developed infrastructure for home automation [2], as well as a nuanced ways of discussing the challenges and opportunities of ubiquitous computing in the home [2,5], the resolution of interaction between features remains largely unexplored. A few HCI studies recognize the challenge and need for resolving the heterogeneity of devices and features [5] and the utility in encapsulating those features into independent functions [4], but we have not explicitly considered the resolution of conflicting functions and features in a complex system with multiple devices. The standard HCI approach of mostly "if-then" scripting (e.g., [6]) for interacting with IoT and home automation leaves the significant tasks of understanding and resolving the resulting feature interactions entirely to the end user. We show that even with a well-specified, verifiable FI resolution mechanism, reasoning about feature interaction is a challenge for most users. Connecting with Software Engineering research can help achieve IoT solutions that are verifiable and scalable [15,22], but achieving these goals in way that match users' mental models is a task that may be particularly well-suited to the skills of HCI researchers. **In this work, we position FI resolution mechanisms as an important area of for future research in IoT and home automation—one that may be particularly well suited to collaboration between SE and HCI investigators.**

ACKNOWLEDGMENTS

With grateful acknowledgement of Eric Cheung's resolution mechanism work, Tom Jenkins' analysis help, Alicia Abella's support, and our participants' time and effort.

REFERENCES

1. A. J. Bernheim Brush, Jaeyeon Jung, Ratul Mahajan, and James Scott. 2012. HomeLab: Shared Infrastructure for Home Technology Field Studies. In *Proc. of Ubicomp*, 1108–1113.
2. A J Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. 2011. Home Automation in the Wild: Challenges and Opportunities. In *Proc. of CHI*, 2115–2124.
3. Muffy Calder, Mario Kolberg, Evan H. Magill, and Stephan Reiff-Marganiec. 2003. Feature interaction: a critical review and considered forecast. *Computer Networks* 41, 1: 115–141.
4. Colin Dixon, Ratul Mahajan, Sharad Agarwal, A. J. Brush, Bongshin Lee, Stefan Saroiu, and Victor Bahl. 2010. The Home Needs an Operating System (and an App Store). In *Proc. of Workshop on Hot Topics in Networks*, 18:1–18:6.
5. W. Keith Edwards and Rebecca E. Grinter. 2001. At Home with Ubiquitous Computing: Seven Challenges. *UbiComp*: 256–272.
6. Justin Huang and Maya Cakmak. 2015. Supporting Mental Model Accuracy in Trigger-action Programming. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '15), 215–225. https://doi.org/10.1145/2750858.2805830
7. Willet Kempton. 1986. Two Theories of Home Heat Control. *Cognitive Science* 10: 75–90.
8. P. Leelaprute, T. Matsuo, T. Tsuchiya, and T. Kikuno. 2008. Detecting Feature Interactions in Home Appliance Networks. In *Proc. of Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 895–903. https://doi.org/10.1109/SNPD.2008.158
9. Sarah Mennicken, Jonas Hofer, Anind Dey, and Elaine M. Huang. 2014. Casalendar: A Temporal Interface for Automated Homes. In *Ext. Abst. of CHI*, 2161–2166.
10. Miji Park, Juhyun Eune, and Suzung Kim. 2010. Tangible User Interface Design for Home Automation Energy Management Appliances. In *Proc. of SIGRAPH Asia*.
11. Ajitha Rajan, Lydie du Bousquet, Yves Ledru, German Vega, and Jean-Luc Richier. 2010. Assertion-based Test Oracles for Home Automation Systems. In *Proc. of Workshop on Model-Based Methodologies for Pervasive and Embedded Software*, 45–52.
12. J. Rasmussen. 1983. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics* SMC-13, 3: 257–266. https://doi.org/10.1109/TSMC.1983.6313160
13. Michael Rietzler, Julia Greim, Marcel Walch, Florian Schaub, Björn Wiedersheim, and Michael Weber. 2013. homeBLOX: Introducing Process-driven Home Automation. In *Proc. of Ubicomp 2013*, 801–808.
14. Irving Seidman. 1998. *Interviewing as Qualitative Research: A Guide for Researchers in Education And the Social Sciences*. Teachers College Press, New York.
15. C. Soares, R.S. Moreira, R. Morla, J. Torres, and P. Sobral. 2012. Prognostic of feature interactions between independently developed pervasive systems. In *Proc. of Conference on Prognostics and Health Management*. https://doi.org/10.1109/ICPHM.2012.6299523
16. C. Soares, R.S. Moreira, R. Morla, J. Torres, and P. Sobral. 2012. Graph-based approach for interference free integration of pervasive applications. In *Proc. of Symposium on Wireless and Pervasive Computing (ISWPC)*. https://doi.org/10.1109/ISWPC.2012.6263661
17. Leila Takayama, Caroline Pantofaru, David Robson, Bianca Soto, and Michael Barry. 2012. Making Technology Homey: Finding Sources of Satisfaction and Meaning in Home Automation. In *Proc. of Ubicomp*, 511–520.
18. Michael Wilson, Mario Kolberg, and Evan H. Magill. 2008. Considering Side Effects in Service Interactions in Home Automation – an Online Approach. In *Feature Interactions in Software and Communication Systems IX*. IOS Press, 172–187.
19. Rayoung Yang and Mark W. Newman. 2013. Learning from a Learning Thermostat: Lessons for Intelligent Systems for the Home. In *Proc. of Ubicomp*, 93–102.
20. Pamela Zave. 2011. Mid-call, Multi-party, and Multi-device Telecommunication Features and Their Interactions. In *Proc. of IPTcomm*, 17:1–17:12.
21. Pamela Zave and Eric Cheung. 2014. *A resolution mechanism for feature interactions in ubiquitous computing*. AT&T Research Labs.
22. Pamela Zave, Eric Cheung, and Svetlana Yarosh. 2015. Toward user-centric feature composition for the Internet of Things. *arXiv:1510.06714 [cs]*. Retrieved December 21, 2016 from http://arxiv.org/abs/1510.06714
23. AT&T Home Security and Automation. Retrieved December 21, 2016 from https://my-digitallife.att.com/learn/home-security-and-automation