Why Can't Smart Phones be Polite, too? What Would a Phone Need to Know?

by

Chih-yu Chao

B.S., National Taiwan University (2000)
M.S., National Taiwan University (2002)
M.A., Carnegie Mellon University (2003)
M.S., Carnegie Mellon University (2005)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2011

Author ....
Department of Electrical Engineering and Computer Science
January 26, 2011

Certified by.......................................
Randall Davis
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by.......................................
Terry P. Orlando
Chair, Department Committee on Graduate Students

Why Can't Smart Phones be Polite, too? What Would a Phone Need to Know?

by

Chih-yu Chao

Submitted to the Department of Electrical Engineering and Computer Science
on January 26, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Mobile devices nowadays contain state-of-the-art technologies and are considered "smart". However, we and others around us are often interrupted or embarrassed by these smart devices because the calls and messages received by the devices are not always presented to us at the right moment with the appropriate modality. Our work investigates what information a device like this needs to know, and how the device should make use of such information in order to behave "politely".

We began by investigating the human definition of "politeness" in the context of handling voice calls and text messages, and we found the common properties shared by the scenarios where a device is expected to behave politely. Next, we built a rule-based decision-making system that infers user interruptability and decides when and how the device should interrupt the user. We then determined whether the vocabulary defined in our rule set has captured general users' definition of a polite device. We also determined that users were able to understand the system's vocabulary and customize the rule set for their own needs. To further accommodate individual users' needs, we created a debugging interface that allows users to explore the rule set and modify the rules when the device "misbehaves". After that, we identified two major challenges in debugging: user's willingness to debug, displaying the structure of the rule set on a small screen real estate. Lastly, we pointed out the aspects that can be investigated in the future to improve our current work, including: augmenting the vocabulary when more signals become available, considering users of different use habits and cultural backgrounds, and designing a better interface that addresses the challenges in debugging.

Thesis Supervisor: Randall Davis
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I would like to thank my thesis advisor Randall Davis, for supporting me intellectually and financially, and for being a great mentor in all aspects. Randy helped me identify the core issues to be addressed in my research, prevented me from being sidetracked by trivial tasks, and patiently reminded me to always answer the right questions. I also thank Randy for his guidance and encouragement outside the research setting, and for generously sharing his life's wisdom.

I am fortunate to have Peter Szolovits and Chris Schmandt on my thesis committee. Pete and Chris offered insightful comments and invaluable suggestions to guide my research to the right direction. I truly appreciate their genuine enthusiasm and immense knowledge in the field, as well as their timely feedback on my work.

I would also like to thank Tim Paek at Microsoft Research for inspiring me to work on a topic that eventually evolved into this thesis.

Much of this work couldn't have been done without the assistance from Niranjala Manokharan and Janet Fischer: Nira took care of all the administrative details to make sure I was well-fed and getting my user studies conducted smoothly. Janet had been helping me deal with various paperwork and deadlines since day one; I'd like to thank her for being always available and for coming up with the best solutions whenever I had questions. Besides, I would like to thank Marcia Davidson, Britton "Bryt" Bradley, Donna Kaufman, and Lisa Bella for being so resourceful and for assisting me whenever I was in need.

Thank you, my dear labmates in the Multimodal Understanding Group: Aaron Adler, Sonya Cates, Andrew Correa, Jacob Eisenstein, Tom Ouyang, Jeremy Scott, Yale Song, Ying Yin, James Oleinik, Kun-yi Chen, ChiuHao Chen, Season Lee, and Simon Xiao. I treasure all the intellectual discussions, interesting stories, funny jokes, and the good times we shared. The wonderful experience I had in the group will remain vividly in my heart whenever I think of my days at MIT!

I would like to thank the faculty members in the Spoken Language Systems Group – Stephanie Seneff, Jim Glass, and Victor Zue – for helping me with a smooth transition during my first few semesters at MIT CSAIL. Also, it was my great pleasure to have worked with the rest of the SLS Group: Chao Wang, Hung-An Chang, Ghinwa Choueiter, Ed Filisko, Alex Gruenstein, Paul Hsu, Ian McGraw, Chia-ying Jacqueline Lee, John Lee, JingJing Liu, Alex Park, Mitch Peabody, Tara Sainath, Ken Schutte, Han Shu, and Yushi Xu.

My life at MIT could not have been so memorable without the company of my friends, including, among others: Cy Chan, Harr Chen, Victor Costan, James Cowling, Harold Fox, Christina Hruska, Szymon Jakubczak, Brendan Juba, Gregory "Gremio" Marton, Ali Mohammad, Bhavna Muthangi, Rafael Nonato, Krzysztof Onak, Pedro Pinto, Jennifer Roberts, Benjamin Rossman, Guy Rothblum, Michelle Sander, Jacob Scott, Oshani Seneviratne, Emily Shen, Yuan Shen, Fuming Shih, Nabeel Siddiqui, Benjamin Snyder, David Sontag, Jim Sukha, Max Van Kleek, Robert Wang, Grace Woo, Helen You, Chen-Hsiang Jones Yu, and Yang Zhang.

Thanks to the EECS minor requirement, I was able to study with course 21M faculty members: Mark Harvey, Don Byron, Donal Fox, Fred Harris, and Keala Kaumeheiwa, while keeping my life balanced by performing with talented musicians at the MIT Jazz Combo, MIT Festival Jazz Ensemble, and MIT Improvisation Ensemble. At the same time, I'd like to thank everyone in the following organizations for broadening my views and for supporting me in many different ways: CSAIL Graduate Student Lunch, MIT Tango Club, MIT Republic of China Student Association, and the Federation of Taiwanese Student Associations in New England.

I owe special thanks to the friendship of Kai-fan Cheng, Jesse Irons, Yi-san Lai, Angelina Lee, Yi-chien Liu, and Vivian Tsai. Thank you for understanding me, trusting me, lifting me up when I was down, and believing in me more than I believed in myself.

Most importantly, I would like to thank my family for their constant love and support for every step of my journey, especially my mother, who has sacrificed so much of herself to give me the best of everything – I could not have achieved this without you, and I am eternally grateful.

Contents

1 Introduction **15**

1.1 Motivation 15

1.1.1 Example Scenario 16

1.2 Summary 17

2 The "Politeness" of a Smart Mobile Device **21**

2.1 User Study: Calendar Data Collection 21

2.2 What Can a Mobile Device Do? 23

2.3 What Does a Mobile Device Need to Know? 24

3 Steps Towards Constructing a Polite Mobile Device **33**

3.1 The Decision-making System 33

3.1.1 Rule Creation 34

3.1.2 Example Scenario 39

3.1.3 User Study: Vocabulary Collection 49

3.1.4 User Study: Rule Understanding Study 63

3.1.5 User Study: Rule Writing Study 71

3.2 Building a Knowledge Base 77

3.2.1 Example Scenario 78

3.2.2 User Study: Debugging Interface Study 86

4 Discussion **89**

4.1 Limitations 90

4.2 Simulated *vs.* Real Signals 91
4.3 User Interface 91

5 Related Work **93**
5.1 Context-Aware Computing 93
5.2 Interruptability 95
5.2.1 Coordinating Interruptions 95
5.2.2 Measuring User Interruptability 97
5.2.3 Existing Applications 100
5.3 Privacy and Social Psychology 101

6 Conclusions and Future Work **103**
6.1 Conclusions 103
6.2 Future Work 105
6.2.1 When more signals on the wishlist are granted 105
6.2.2 User demographics 105
6.2.3 A better debugging interface 106

List of Figures

2-1 A portion of the calendar data from a subject 32

3-1 Tree structure, illustrating the first rule. 41

3-2 The first 4 conditions of the first rule hold, and hence the first 4 leaf nodes are marked with checks. 41

3-3 The first condition holds, and hence the first leaf node is marked with a check. 42

3-4 The rule in Step 3 is fired with strength of 0.9, which is the strength of the condition of the rule in Step 2. 42

3-5 The rule in Step 2 is fired with strength of 0.72, which is the strength of the 5^{th} condition of the first rule. 43

3-6 The rule in Step 5 is fired with strength of 0.576, which is the strength of the 6^{th} condition of the first rule. 44

3-7 The rule in Step 6 is fired with strength of 0.576, which is the strength of the 7^{th} condition of the first rule. 45

3-8 The rule in Step 7 is fired with strength of 0.504, which is the strength of the 8^{th} condition of the first rule. 46

3-9 The rule in Step 8 is fired with strength of 0.504, which is the strength of the 9^{th} condition of the first rule. 47

3-10 The weakest strength of belief in the conditions is 0.504; the rule is fired with a strength of $0.504 * 1.0 = 0.504$. 48

3-11 Number of subjects considering the phone misbehaving in each scenario. 50

3-12 Vocabulary used by the subjects to describe why the phone misbehaved in the "class" scenario. 52

3-13 Vocabulary used by the subjects to describe why the phone misbehaved in the "movie" scenario. 53

3-14 Vocabulary used by the subjects to describe why the phone misbehaved in the "meeting" scenario. 54

3-15 Vocabulary used by the subjects to describe why the phone misbehaved in the "long bus trip" scenario. 55

3-16 Vocabulary used by the subjects to describe why the phone misbehaved in the "biking" scenario. 55

3-17 Vocabulary used by the subjects to describe why the phone misbehaved in the "sleeping" scenario. 56

3-18 Vocabulary used by the subjects to describe how they would teach the phone to behave in the "class" scenario. 58

3-19 Vocabulary used by the subjects to describe how they would teach the phone to behave in the "movie" scenario. 59

3-20 Vocabulary used by the subjects to describe how they would teach the phone to behave in the "meeting" scenario. 60

3-21 Vocabulary used by the subjects to describe how they would teach the phone to behave in the "long bus trip" scenario. 61

3-22 Vocabulary used by the subjects to describe how they would teach the phone to behave in the "sleeping" scenario. 62

3-23 Number of participants of the rule understanding study 66

3-24 Matching subject's interpretation against the meaning of the rule. Text highlighted in blue indicates a match; text highlighted in red indicates a mismatch. 66

3-25 Percentage of subjects who understood a rule perfectly 70

3-26 Percentage of subjects unable to understand a rule at all 70

3-27 Subject's rule and the existing rule for the "in class" scenario. The condition in blue indicates a match in vocabulary, while the condition in red indicates a mismatch. 75
3-28 Averaged percentage of vocabulary overlap for rules produced by programmers. 76
3-29 Averaged percentage of vocabulary overlap for rules produced by non-programmers. 76
3-30 The interface displays a list of call and message histories. 79
3-31 The user selects an instance where the device was not expected to ring. 79
3-32 The interface displays the information available to the device when the misbehavior took place. 80
3-33 The user wants to see rules that might have caused the device to ring. 80
3-34 The interface displays the rule numbers of rules related to ringing. . . 81
3-35 The user is able to view the content of the rules by clicking on the rule numbers. 81
3-36 The rule selected (#17) was not applied to the situation, because some conditions did not hold. 82
3-37 The rule selected (#74) was applied to the situation. 82
3-38 The user explores further to see why the system thought the scenario was "work". 83
3-39 The rule selected (#73) was applied to the situation because the condition held. 83
3-40 The user decides to add a condition to make the rule less general. . . 84
3-41 The user selects the attribute "`number_of_companions`" from the pull-down menu. 84
3-42 The user completes the new condition: "`number_of_companions` >2". 85
3-43 The rule has been modified with the new condition "`number_of_companions` >2". 85
3-44 The modified rule is saved in the system. 86

List of Tables

2.1 Signals and inferred information related to "environment" that would be good for a mobile device to know in order to behave politely. . . . 26

2.2 Signals and inferred information related to "device" that would be good for a mobile device to know in order to behave politely. 27

2.3 Signals and inferred information related to "presence" that would be good for a mobile device to know in order to behave politely. 27

2.4 Signals and inferred information related to "physical information about the user" that would be good for a mobile device to know in order to behave politely. 28

2.5 Signals and inferred information related to "motion" that would be good for a mobile device to know in order to behave politely. 28

2.6 Signals and inferred information related to "activity" that would be good for a mobile device to know in order to behave politely. 29

2.7 Signals and inferred information related to "time" that would be good for a mobile device to know in order to behave politely. 29

2.8 Signals and inferred information related to "user-perceived control" that would be good for a mobile device to know in order to behave politely. 29

2.9 Signals and inferred information related to "user preference" that would be good for a mobile device to know in order to behave politely. . . . 30

2.10 Signals and inferred information related to "nature of interruption" that would be good for a mobile device to know in order to behave politely. 30

3.1 Vocabulary for what the device can do 35
3.2 Vocabulary for what signals the mobile device knows 36
3.3 Vocabulary for what signals the mobile device knows, continued . . . 37
3.4 Vocabulary for what information the mobile device can infer 38

Chapter 1

Introduction

1.1 Motivation

With the popular use and ubiquity of mobile devices, various technologies have been applied to make these devices "smarter". However, whenever a mobile device, containing advanced technologies, receives incoming information (voice or text) that addresses its user, the device typically initiates an interaction by ringing or vibrating, requesting the user's full attention. As a result, the user is always interrupted regardless of their current activity. If the current activity requires that the user be fully engaged with others, or that all participants pay full attention, the interruption caused by the mobile device will typically be seen as unfavorable or socially unacceptable.

Attempts have been made to make mobile devices more polite and less distracting. For example, Hinckley and Horvitz [12] prototyped a mobile phone that could choose a notification modality with less attentional demand, based on user's initial reaction to the audio/visual alert. For example, if the phone rings and the user is not ready to answer it, the user can touch the phone to lower the volume of the ringer; if the user is holding the phone, the phone will vibrate (instead of ringing) when there is an incoming call. Yet such a device still interrupts its user in the first place when a phone call is received.

As humans, we can easily find out whether someone is busy or available by observing various visual and aural cues. Alternatively, we ask politely for the person's attention, or choose another interaction modality in a manner appropriate to the situation at hand [35]. With the advancements in technology, we would hope for the same from a "smart" mobile device; that is, it ought to be as polite as a human in terms of handling interruptions, and being able to intelligently determine whether to interrupt its user upon receiving an incoming call or message.

1.1.1 Example Scenario

Suppose the user is at the movies, and a voice call is received from the user's coworker. In our work, the smart mobile device knows the following facts from its sensors and access to the user's electronic calendar:

- Caller is a coworker (based on caller ID and user-preset contact list)
- Incoming call type is voice (as opposed to text message)
- Location is "AMC Lowes Boston Common 19" (the tag on the electronic map is acquired by GPS coordinates)
- The user is barely moving (detected by accelerometers)
- The current calendar entry reads "Harry Potter @ AMC"

The decision-making system in the device contains a rule that says if the following conditions hold, the mobile device should tell the caller that the user is busy, and if the caller still wants to talk to the user, the mobile device should vibrate:

1. Caller is a coworker
2. Incoming call type is voice
3. Caller did not indicate that the call is urgent
4. The current calendar entry contains keywords related to performing arts
5. The user will be irritated by aural disruption

6. The user will be irritated by visual disruption
7. Others around the user will be irritated by aural disruption
8. Others around the user will be irritated by visual disruption

Based on the facts known to the mobile device, conditions 1 to 3 hold. The keyword-spotting mechanism in our system recognizes names of movie theaters such as "AMC" and relates them to performing arts, which becomes a known fact to the mobile device. For conditions 5 to 8, our system infers whether each of them is true by using other rules. For example, the system finds a rule stating that condition 5 is true if the user is in a performance venue and barely moving. The system also finds other rules stating that conditions 6, 7, and 8 are true if the user is in a performance venue and barely moving. Since all 8 conditions are true, the mobile device now tells the caller that the user is busy, without interrupting the user. If the caller still indicates the intention to talk to the user, the mobile device will vibrate to inform the user of the incoming call.

1.2 Summary

In order for a mobile device to behave as "politely" as the example illustrated above, first of all we need to determine when it is important for the phone to be polite; that is, what is the human definition of "politeness" with regard to handling voice calls and text messages. Next, we need to determine what the mobile device needs to know, and how the device could make use of the information acquired or inferred in order to behave politely. Finally, we need to recognize that it is impossible for a polite mobile device constructed this way to cover all the situations in life where it is expected to behave politely; it is also impossible for such a device to cater to individual needs and preferences. Hence, we need to provide a means to allow the user to add to the device's store of knowledge about politeness when the device misbehaves.

To address these issues, we started out by collecting real-life scenarios where users think it is important that the mobile device behave politely. We then found the common properties shared by the 30 unique scenarios where most users would find it embarrassing or irritating if a device initiates the interruption with the wrong timing or wrong modality: (1) the user's aural/visual attention is not to be disrupted (2) the aural/visual attention of the bystanders (others around the user) is not to be disrupted (3) the user is not physically available to respond to the call.

Next, we selected the output modalities a mobile device could use in interacting with the user and the caller. We also investigated what would be helpful for the device to know in order to behave politely, and came up with a list of signals and inferred information that were realistic for our work.

Then, we defined the vocabulary for our system based on the scenarios collected and the common properties they share; at the same time, we conducted a user study to look into the vocabulary used by general users to describe the scenarios that require a mobile device to be polite, and the vocabulary of how the users would teach the mobile device to behave politely. The result shows that the vocabulary defined in our system is able to cover 42% of the user vocabulary.

With the intention of making the reasoning process transparent to the user, we chose a rule-based approach for the decision-making system. The system makes inferences of user interruptability based on information available to the device, in order to determine how the device should respond to an incoming call or message. The current rule set used by the system contains 250 rules. Based on our user study, 2/3 of the rules could be understood perfectly by at least half of the programmers without any assistance.

The rule set is meant to be customizable; hence we conducted another user study to find out whether users would be able to create rules using the vocabulary defined in our work. With a minimal amount of introduction to the vocabulary, syntax, and the basics of rule writing, most users were able to create rules with the vocabulary for their own needs. For rules produced by programmers, at least half of the user vocabulary is covered by the existing rule set in most cases.

To make the system more accommodating to individual users' needs and to cope with scenarios that are not covered by the existing rule set, we created a debugging interface for the users to modify existing rules and create new rules. A user study was then conducted to see whether such an interface was usable. When subjects in our user study were told to debug purposefully defective rules to correct the behavior of the device, most of them were able to fix the rules using our debugging interface. However, we noticed two major issues that would prevent users from debugging – user's unawareness of the concept of "debugging", and the difficulty of keeping track of the structure of the rule set on a small screen estate.

There is substantive room for improvement before our current system could become a product released to the general public. The findings from the exploratory studies have helped us point out the issues and questions that shall be addressed in the future (such as user's willingness to debug, presenting the structure of the rule set on a small screen real estate, user demographics, etc.), in hopes of constructing a polite mobile device.

Chapter 2

The "Politeness" of a Smart Mobile Device

Before constructing a polite mobile device, we first need to understand when it is important to have a device that behaves politely. Then we need to determine what output modalities a mobile device could use, in order to behave politely under different circumstances. We also need to investigate what signals and inferred information the device needs to know to behave politely, and decide on a list of signals and inferred information realistic for our work.

2.1 User Study: Calendar Data Collection

In order to find out in what situations users would want a device to behave politely, we recruited 6 subjects (including 4 MIT students) for a data collection study. We asked the subjects to record their daily activities on Google Calendar for 7 consecutive days, in the form of 30-minute log entries without blanks. Each entry in the log contains the activity, location, and participants. Figure 2-1 shows a portion of the calendar data of one subject. Notice that some entries contain acronyms or the initials of participants. Some subjects would put "private" in their entries to indicate activities they were not willing to disclose.

The subjects were interviewed individually to review each log entry, so that they could point out what device behavior would be considered polite or otherwise in different scenarios. Subjects were asked to describe the location of each activity, the physical movement involved, the relation between self and each participant when applicable. (Subjects were not asked to reveal the identity of event participants or the content of the private activities; we were only interested in how such information can be represented with the available information from a mobile device's point of view.) While reviewing each entry, the subjects were asked, "*Suppose you have a smart assistant who knows everything about you and handles your phone calls and messages. If you have an incoming call or text message at this moment, what should your assistant do?*". Then the subjects were asked to explain what the assistant should do and why other possible behaviors would be considered impolite. (For example, "*I don't want to disrupt the meeting, so the assistant should just take the message for me instead of letting my phone go off. I can get back to them after the meeting.*")

Based on the calendar data collection effort, there were about 30 unique scenarios where it is important for a device to behave politely. The scenarios can roughly be categorized as follows:

1. late at night
2. meeting, concert, movie, class, group work
3. working, library, in court, visiting police department, visiting embassy
4. shower, studying, "me time"
5. napping, sleeping, hangover
6. at home with significant other, significant other sleeping
7. lunch/dinner with significant other or with friends, talking to others
8. long bus/train trip
9. short bus/subway ride

10. biking
11. rehearsal
12. clubbing, party, bar

Given the diversity of these scenarios, we decided to go for a general approach to answer the question: *what are the factors that make it important to have a polite device?* With the explanations from the subjects, we were able to extract the common properties shared by the scenarios mentioned above, which are:

- User's aural attention
- User's visual attention
- Others' aural attention
- Others' visual attention
- User's perceived privacy
- User's physical availability (*e.g.* User's hands are too busy to reach for the device.)
- Urgency of incoming information

2.2 What Can a Mobile Device Do?

After understanding the different circumstances where a device is expected to behave politely, we would like to know what output modalities could be used to notify the user under these circumstances. The following is a list of output modalities that can be chosen to notify or present the information to the user:

1. Ring
2. Vibrate

3. Silent
4. Flash
5. Beep
6. Beep2: beep differently to indicate "I need your attention when you get a chance"
7. Reveal-busy: tell the caller that the user seems busy, and let the caller decide whether the phone call should go through
8. Reveal-location: tell the caller where the user seems to be, and let the caller decide whether the phone call should go through
9. Read out loud the content of a text message
10. A combination of the above: for example – reveal-location, if the caller wants the phone call to go through, vibrate and ring

The first 5 modalities already exist in current smart phone models. Beep2 is a more gentle way to request the user's attention. Reveal-busy and reveal-location allow the device to interact with the caller prior to interrupting the user. Reading out loud the content of a text message is helpful when the user is physically unavailable to reach for the device (for example, doing dishes in the kitchen). The option of combining multiple modalities allows the device to notify the user with more levels of intensities than a typical smart phone.

2.3 What Does a Mobile Device Need to Know?

With the fast advancement of technology, we can imagine a future mobile device embedded with all kinds of sensors, and our environment being instrumented with devices that emit or collect various signals. Without limiting ourselves to the mobile technologies available at present, we would like to investigate what would be good for a device to know in order to behave politely.

To simplify our discussion, we define 3 terms:

- Signals: measurements or data that can be acquired directly from a mobile device, using its built-in sensors or electronic calendar entries
- Inferred information: what can be inferred from signals
- Information: a general term to refer to signals and/or inferred information

Information that will be helpful for a mobile device to become polite may be concerned with a variety of elements in the world. For example:

1. Environment
2. Device
3. Presence
4. Physical information about the user
5. Motion
6. Activity
7. Time
8. User-perceived control (over the situation/device)
9. User preference
10. Nature of interruption

In Tables 2.1 to 2.10, we list the signals and inferred information related to each element, along with how the information can be obtained, the tools/technologies required, and an example that signifies the importance of such information. In the tables, the word *phone* is used to refer to 'built-in sensor(s) in a smart mobile device".

Take the first 6 tables for example: In Table 2.1, if the mobile device knows the location of the user via its GPS/WLAN positioning mechanisms, it will know not

ENVIRONMENT			
signal	how to obtain	tool required	example (importance of the information)
location	GPS/WLAN positioning	phone	movie theater *vs.* home
humidity	humidity	phone	taking shower
temperature	temperature	phone	taking shower
ambient acoustical information	echo, reverberation	phone, microphone in room	living room *vs.* stadium
illumination	illumination	light sensor in room, phone (not in pocket)	dark room implies user sleeping
building type	location	knowledge base	residence *vs.* office
room type	location	knowledge base	seminar room, restroom, office
particular location in a room	location	pressure/light sensor in room	cooking in the kitchen
vehicle type	location, vehicle ID	knowledge base, sensor in vehicle	car, bus, train

Table 2.1: Signals and inferred information related to "environment" that would be good for a mobile device to know in order to behave politely.

to make a sound when the location is a movie theater. In Table 2.2, if the device knows it is being flipped over, it is likely that the user does not want to be disturbed. In Table 2.3, if the device is able to detect the number of speakers, it will know not to make a sound when there is only one person speaking. In Table 2.4, if the device recognizes the hand gesture made by the user, then it knows not to interrupt the user when the user is waving his hand. In Table 2.5, if the device detects that the user suddenly accelerates, it is likely that the user is in a rush and cannot take any incoming calls. In Table 2.6, if the device has access to the user's calendar and understands the content of the entries, it will know not to interrupt the user when the current calendar entry says "mid-term exam".

In our work, we do not intend to instrument the environment with sensors or any devices; nor do we intend to attach objects to the user's body or to build a knowledge base. With this in mind, the list of signals considered available to the mobile device in our work is only a subset of those listed in Tables 2.1 to 2.10. To be more specific,

DEVICE			
signal	how to obtain	tool required	example (importance of the information)
connection to/of other devices	device connection	phone, signal-sniffing instrument in room	laptop-projector connection implies a presentation
motion of device	speed, acceleration	phone	phone being flipped over (gesture for "do not disturb"), user in a vehicle

Table 2.2: Signals and inferred information related to "device" that would be good for a mobile device to know in order to behave politely.

PRESENCE			
signal	how to obtain	tool required	example (importance of the information)
number of speakers	voice(s) of speaker(s)	phone (with speaker identification/diarization)	before *vs.* during a lecture
number of others present	number of other personal devices	phone (with bluetooth), pressure sensor on chair	group *vs.* individual meeting
speaker identity	voice or face image of speaker	phone (with speaker identification), camera in room (with face identification)	supervisor *vs.* family member

Table 2.3: Signals and inferred information related to "presence" that would be good for a mobile device to know in order to behave politely.

PHYSICAL INFORMATION ABOUT THE USER			
signal	how to obtain	tool required	example (importance of the information)
body temperature	body temperature	phone	working out *vs.* asleep
blood pressure	blood pressure	phone	working out *vs.* asleep
heart rate	heart rate	phone	working out *vs.* asleep
gesture, gaze, eye contact avoidance	line of sight, gesture	camera in room (gesture understanding required), phone (not in pocket)	"give me 5 minutes"

Table 2.4: Signals and inferred information related to "physical information about the user" that would be good for a mobile device to know in order to behave politely.

MOTION			
signal	how to obtain	tool required	example (importance of the information)
user motion (walking, running, sitting, etc.)	acceleration	phone, motion sensor in room	running implies being in a rush
current motion of others	acceleration	motion sensor in room	audience sitting still *vs.* standing in a concert
duration of no active user motion	acceleration, timestamp	phone, motion sensor in room	lying down implies sleeping
transition between user motions	acceleration	phone, motion sensor in room	standing up after lecture is over

Table 2.5: Signals and inferred information related to "motion" that would be good for a mobile device to know in order to behave politely.

ACTIVITY			
signal	how to obtain	tool required	example (importance of the information)
user interaction with software application	which application is on top; what else are open	known information	Eclipse *vs.* Solitaire
current task that the user is attending to	planned events on calendar	text understanding	movie, exam
physical objects currently being used/touched	physical objects currently being used/touched	sensor on object	steering wheel, machinery, flask, spatula

Table 2.6: Signals and inferred information related to "activity" that would be good for a mobile device to know in order to behave politely.

TIME			
signal	how to obtain	tool required	example (importance of the information)
day of week, time of day	day of week, time of day	known information	3:00am implies sleeping
time until the next (planned) event takes place	planned events on calendar	text understanding	20 minutes before exam

Table 2.7: Signals and inferred information related to "time" that would be good for a mobile device to know in order to behave politely.

USER-PERCEIVED CONTROL (OVER THE SITUATION/DEVICE)			
signal	how to obtain	tool required	example (importance of the information)
door openness	door openness	sensor in room	closed door implies "do not disturb"
window opaqueness	window opaqueness	sensor in room	closed blinds implies "privacy please"
crowdedness (density)	number of people/area	sensor in room, knowledge base	packed bus *vs.* alcove
psychological control: freedom of deciding with whom to share personal information	identity of bystander	inference from calendar entry, camera in room (with face identification)	significant other *vs.* unknown bystanders

Table 2.8: Signals and inferred information related to "user-perceived control" that would be good for a mobile device to know in order to behave politely.

USER PREFERENCE			
signal	how to obtain	tool required	example (importance of the information)
a fixed period of time (specified by the user) where no interruption is allowed	time specified by the user	known information	during exam

Table 2.9: Signals and inferred information related to "user preference" that would be good for a mobile device to know in order to behave politely.

NATURE OF INTERRUPTION			
signal	how to obtain	tool required	example (importance of the information)
nature of the incoming information	email, phone call, IM, text message	known information	text messages are perceived as more private than normal calls or emails; calls may be more urgent than IM
sender	sender name and/or email address, caller ID	known information	(self-explanatory)
number of recipients	number of recipients	known information	cc'ed, unclosed-recipients
nature of sender and recipients	nature of sender and recipients	knowledge base	supervisor *vs.* friend
case of email	uppercase / lowercase	known information	*URGENT!!!*
content of the header and body	content of the header and body	inference (natural language understanding required)	family emergency *vs.* spam

Table 2.10: Signals and inferred information related to "nature of interruption" that would be good for a mobile device to know in order to behave politely.

the following signals were used in our system:

1. Environment: location (GPS, IP address), ambient noise, vehicle type
2. Device: moving pattern of the device (*e.g.* being flipped over)
3. Presence: number of companions, other device detected
4. Motion: moving speed and moving pattern of the user (*e.g.* running)
5. Activity: user's calendar entries
6. Time: day of week, time of day
7. Nature of interruption: nature of incoming information (voice call *vs.* text message), sender, sender's intention

In addition, our system makes inferences about the following information using the signals available:

- Location (calendar)
- Bystanders' ears/eyes busy
- Bystanders irritated by aural/visual disruption
- User's ears/eyes busy
- User irritated by aural/visual disruption
- User's hands busy
- User's perceived control over privacy

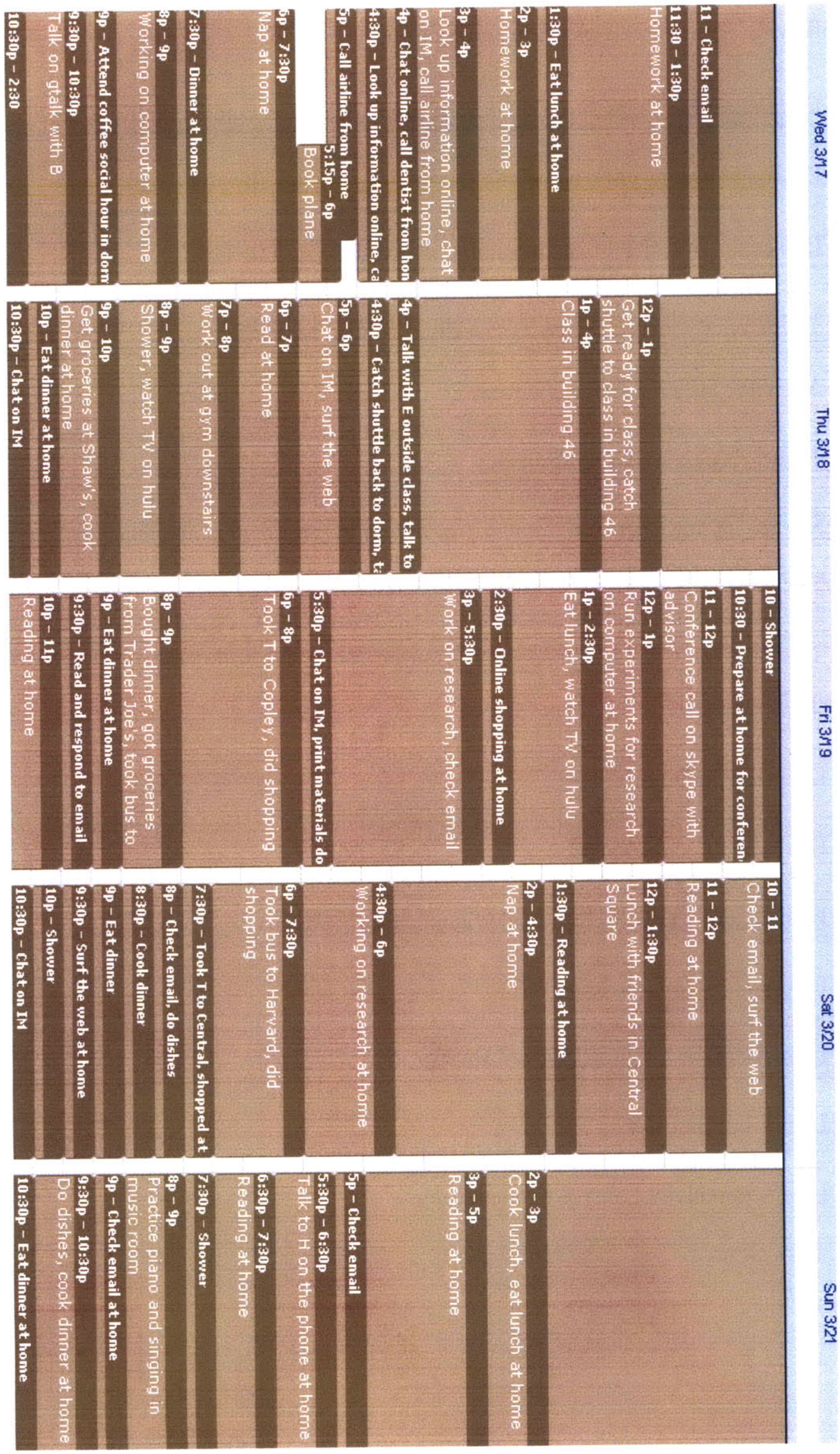

Figure 2-1: A portion of the calendar data from a subject

Chapter 3

Steps Towards Constructing a Polite Mobile Device

After investigating (1) when it is important for a mobile device to be polite, (2) what the device can do to respond to incoming calls and messages, and (3) what information it needs to know in order to behave politely, we would like to focus on the next question: How does a mobile device make use the information available to determine when and how to interrupt the user?

3.1 The Decision-making System

To make decisions about when and how to interrupt the user, we used a rule-based approach [8] to keep the reasoning process transparent to the user. We defined the vocabulary for the rules based on available signals and the result of a data collection study (described in 2.1), and created rules that help the device make use of the information available to determine how to behave.

A rule-based decision-making system requires expert knowledge to build the rules. In our case, users of a mobile device know when they do not want to be interrupted; they also know in what situations they would feel embarrassed or irritated if a device suddenly rings, speaks, vibrates, or flashes. The users are able to describe the reasons why it is appropriate (or otherwise) for a device to go off at a given moment (further described in Section 3.1.3). In other words, every user of a mobile device qualifies as an "expert" in this particular domain. The rules written for the decision-making system will not describe the entire universe; instead, these rules are to describe how a device should behave, simply from the scope of the device itself (based on the information collected, measured, or inferred).

3.1.1 Rule Creation

As humans, when we make a decision on whether to interrupt someone, we tend to first consider the scenario the person is in (for example, having a meeting with a supervisor, making an important client presentation, etc.), and then to consider how to interact with the person in order to minimize the cost of interruption. We could choose to create rules to describe each scenario, and map the scenario to responses that are considered polite; however, it is impossible to enumerate all possible scenarios in real life. There are several fundamental properties that many scenarios share (as described in 2.1), and hence a device only needs to learn about these fundamental properties to determine how to behave. To be more concrete, if a user does not want his device to ring when he is meeting with his supervisor, or when he is making an important presentation in front of clients, then his device should have a rule that says: `whenever it is during regular work hours, the user is surrounded by at least one person in a quiet environment, and both the user and others around him will be irritated by aural disruption, then remain silent`. Note that "time of day is during regular work hours", "there is at least one companion", "the environment is quiet", "user will be irritated by aural disruption", and "others around the user will be irritated by aural disruption" are the properties shared by these scenarios, and that the rule above given to the device also applies to the scenario where the user is sitting in a lecture. Without having to predict whether the user is actively or passively participating in a meeting or whom the user is with, the device only needs to know the facts stated in the rule in order to behave politely.

Using this idea, we came up with around 250 rules that cover the 30 unique scenarios from the calendar data collection effort (Section 2.1). The end of Chapter

2 has described how we defined the vocabulary used in the rules, listed in Tables 3.1 to 3.4.

WHAT THE DEVICE CAN DO		
Attribute	Operators	Values

`response`	`=, !=`
	`is, is not`
		`flash`
		`ring`
		`beep`
		`beep2`
		`vibrate`
		`reveal-busy`
		`reveal-location`
		`silent`
		`read message content out loud`
		`any combination of the above`

Table 3.1: Vocabulary for what the device can do

WHAT SIGNALS THE MOBILE DEVICE KNOWS		
Attribute	Operators	Values
ambient noise	>, <, = greater than, less than, equals	any number (dB)
call type	= is	voice sms
caller	=, != is, is not	important caller expected caller boss coworker rejected caller any user-defined categories any name in the contact list
current calendar entry	= is	''any text'' caller class meal meeting party performing arts significant other spots friend any combination of the above
location (GPS)	= is	tagged location name
moving pattern	=, != is, is not	phone immobile face down phone immobile face up user running user sitting user walking

Table 3.2: Vocabulary for what signals the mobile device knows

WHAT SIGNALS THE MOBILE DEVICE KNOWS (CONT.)		
Attribute	Operators	Values
moving speed mph	>, <, = greater than, less than, equals	any non-negative number (mph)
number of companions	>, <, = greater than, less than, equals	any non-negative integer
other device detected	= is	bluetooth headset landline phone other cell phone tv projector
physical location (IP address, WLAN location)	= is	classroom embassy gym library public vehicle home restaurant
time of day	= is	between(start_time, end_time)
urgency	=, != is, is not	urgent work-related

Table 3.3: Vocabulary for what signals the mobile device knows, continued

WHAT THE DEVICE CAN INFER		
Attribute	Operators	Values
bystanders ears busy	= is	true false
bystanders eyes busy	= is	true false
bystanders irritated by aural disruption	= is	true false
bystanders irritated by visual disruption	= is	true false
hands busy	= is	true false
location	= is	performance venue public location quiet public location private location tagged location name
my ears busy	= is	true false
my eyes busy	= is	true false
me irritated by aural disruption	= is	true false
me irritated by visual disruption	= is	true false
perceived control over privacy	= is	high low medium

Table 3.4: Vocabulary for what information the mobile device can infer

3.1.2 Example Scenario

To demonstrate how our rule-based system works, we use the same example scenario as in Chapter 1: the user is at the movies, and a voice call is received from the user's coworker.

Scenario: user is at the movies, call is received from user's coworker

When the user is at the movies and there is an incoming call, it is somewhat impolite to leave the seat to take the call. However, if the call is from someone important and the message is urgent, the user still wants to be notified immediately and to have the chance to take the call. When the incoming call type is in the form of a voice call, we do not require the mobile device to predict its urgency; instead, the device would reveal the user's availability state and leave the decision to the caller. If the caller decides to interrupt the user, it would be polite for the device to notify the user with a less intrusive modality (for example, vibration).

We are going to show a set of rules that a device uses to determine how to react in such scenario. Suppose the device knows the following facts:

- Caller is a coworker (based on caller ID and user-preset contact list)
- Incoming call type is voice
- Location is "AMC Loews Boston Common 19" (the tag on the electronic map is acquired by GPS coordinates)
- Moving pattern detected is user sitting
- Moving speed is 0.01 mph
- The current calendar entry reads "Harry Potter @ AMC"

In the rule set, each rule starts with the conclusion (indicated by the `=>` symbol), followed by one or more conditions. All the conditions are connected with `AND` by default. The number next to the conclusion indicates how strongly the system should believe in such conclusion.

The decision-making system first searches the rule set for all the rules that contain the attribute `response` in the conclusion (*i.e.* rules that suggest how the device should respond). After examining whether each condition in the rules holds, and how strongly each conclusion can be believed, the system selects the conclusion with the highest strength of belief and makes the device behave as described in the conclusion.

In this example, we start by selecting one of these rules to demonstrate the reasoning process of the decision-making system:

```
=> response = (reveal-busy, vibrate) 1.0
call_type = voice
urgency != urgent
caller = coworker
current_calendar_entry = performing_arts
perceived_control_over_privacy = low
me_irritated_by_aural_disruption = true
me_irritated_by_visual_disruption = true
bystanders_irritated_by_aural_disruption = true
bystanders_irritated_by_visual_disruption = true
```

The first rule says that if there is a non-urgent voice call from a coworker, the user's current calendar entry contains words associated with performing arts, the user seems to have low control over his privacy, the user will be irritated by aural and visual disruptions, and others around the user will be irritated by aural and visual disruptions, then the device should believe with full (1.0) confidence that it is right to respond to the voice call by telling the caller that the user is busy; if the caller still wants to talk to the user, then the mobile device should vibrate.

We can build a tree structure, with the conclusion being the root node, and the conditions being the leaf nodes (as in Figure 3-1).

Step 1 – Based on the information it already has, the phone knows that the 1^{st} and 3^{rd} conditions hold. The caller did not indicate that the call was urgent, so the 2^{nd} condition holds. The 4^{th} condition holds because the keyword spotting mechanism

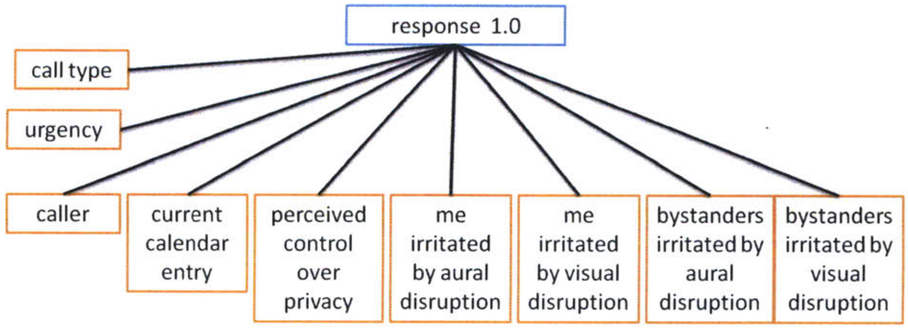

Figure 3-1: Tree structure, illustrating the first rule.

in our system recognizes the name of the movie theater (AMC) and associates it with the concept of performing arts. Now the system needs to find out whether the rest of the conditions hold by exploring rules related to these conditions. (See Figure 3-2.)

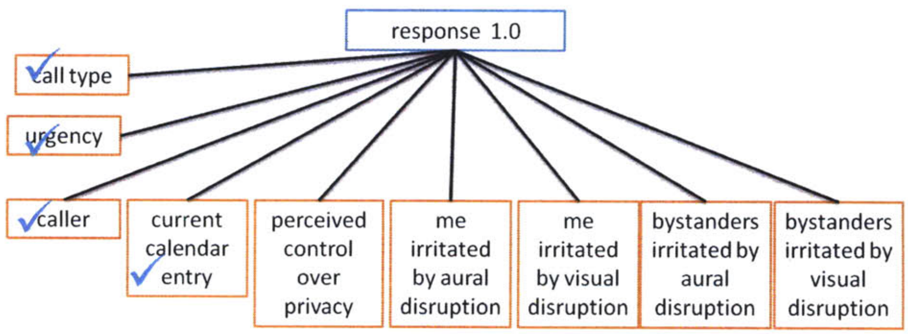

Figure 3-2: The first 4 conditions of the first rule hold, and hence the first 4 leaf nodes are marked with checks.

Step 2 – The system finds a rule:

```
=> perceived_control_over_privacy = low 0.8
moving_pattern = user_sitting
location = performance_venue
```

which says that if the moving pattern indicates that the user is sitting, and the user's location is a performance venue, then the system should believe with 0.8 (80%) strength that the user has low control over his privacy. The first condition holds based on the information known to the device. Now the system needs to find out whether the second condition holds by exploring the rules set. (See Figure 3-3.)

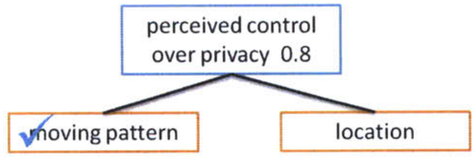

Figure 3-3: The first condition holds, and hence the first leaf node is marked with a check.

Step 3 – The system finds a rule:

```
=> location = performance_venue 0.9
current_calendar_entry = performing_arts
```

which says that if the user's current calendar entry contains words associated with performing arts, then the system should believe with 0.9 strength that the user's location is a performance venue. The condition in this rule holds, and thus the rule is fired with 0.9 strength. This means that the second condition of the rule in Step 2 holds, with a strength of 0.9. (See Figure 3-4.)

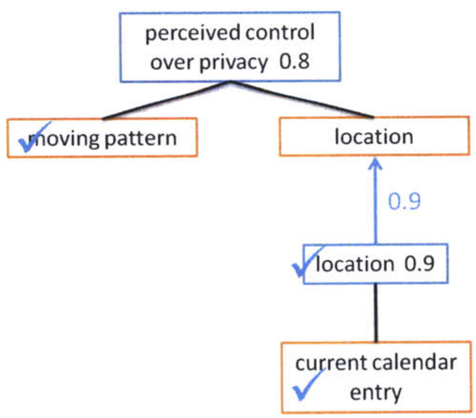

Figure 3-4: The rule in Step 3 is fired with strength of 0.9, which is the strength of the condition of the rule in Step 2.

Step 4 – Both conditions of the rule in Step 2 hold; hence the rule is fired with a strength of $min(1.0, 0.9) * 0.8 = 0.72$. This means that the 5^{th} condition of the rule in Step 1 holds, with a strength of 0.72. (See Figure 3-5.)

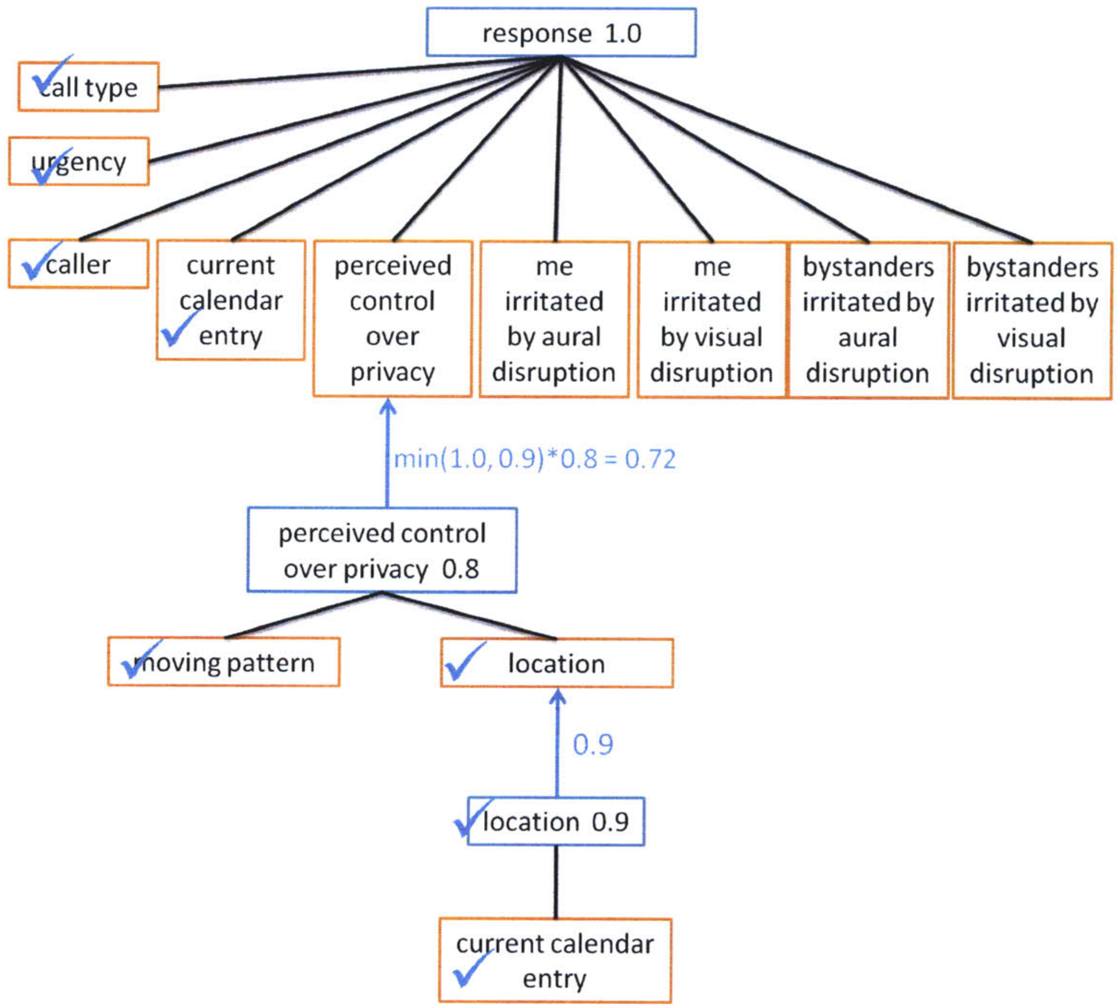

Figure 3-5: The rule in Step 2 is fired with strength of 0.72, which is the strength of the 5$^{\text{th}}$ condition of the first rule.

Step 5 – To figure out whether the 6$^{\text{th}}$ condition of the rule in Step 1 holds, the system finds a rule:

```
=> me_irritated_by_aural_disruption = true 0.8
perceived_control_over_privacy = low
location = performance_venue
```

which says that if the user seems to have low control over his privacy, and the user's location is a performance venue, then the system should believe with 0.8 strength that the user will be irritated by aural disruption. The first condition holds due to Step 4, and the second condition holds due to Step 3. Hence the rule is fired with a strength of $min(0.72, 0.9) * 0.8 = 0.576$. And the 6^{th} condition of the rule in Step 1 holds, with a strength of 0.576. (See Figure 3-6.)

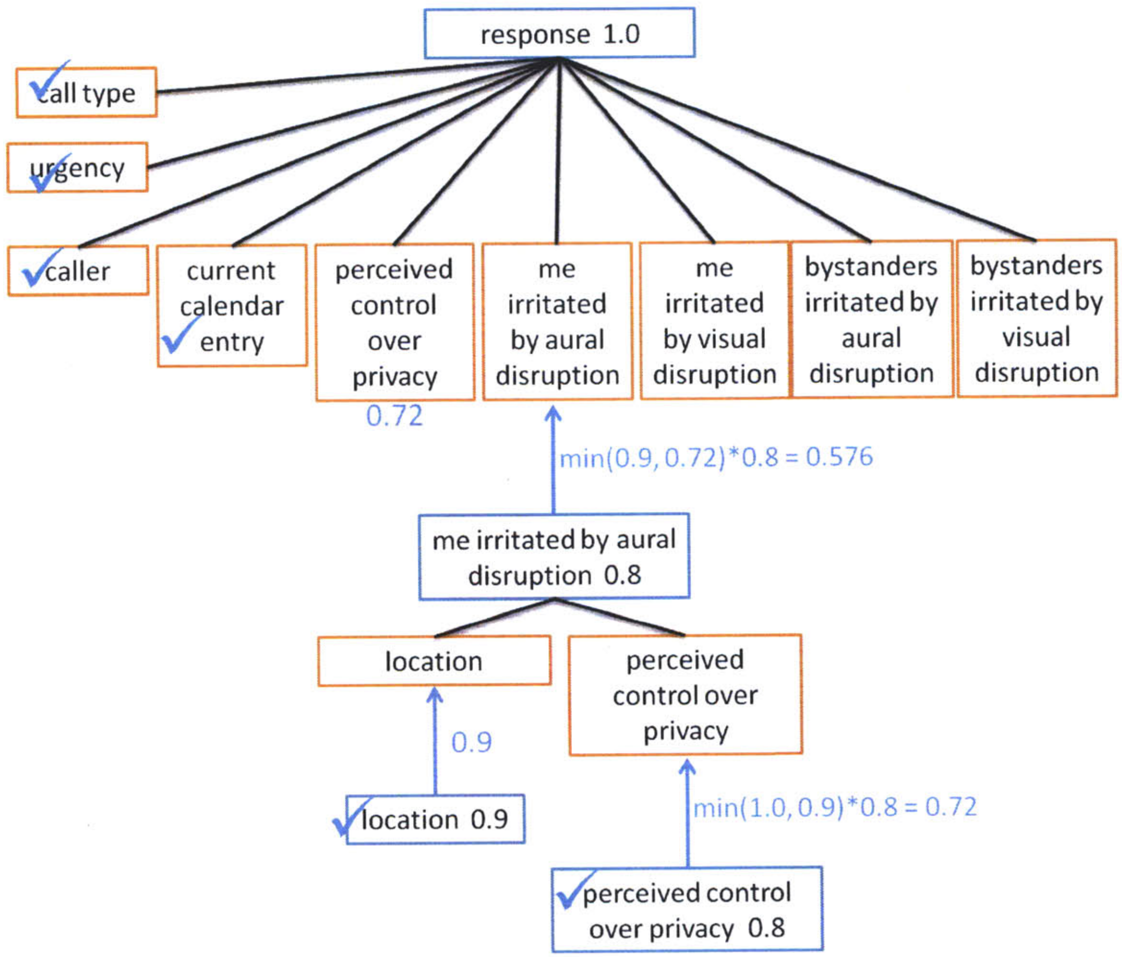

Figure 3-6: The rule in Step 5 is fired with strength of 0.576, which is the strength of the 6^{th} condition of the first rule.

Step 6 – To figure out whether the 7^{th} condition of the rule in Step 1 holds, the system finds a rule:

```
=> me_irritated_by_visual_disruption = true 0.8
perceived_control_over_privacy = low
location = performance_venue
```

which says that if the user seems to have low control over his privacy, and the user's location is a performance venue, then the system should believe with 0.8 strength that the user will be irritated by visual disruption. This rule shares the same conditions as the previous rule, and hence the rule is fired, and the 7th condition of the rule in Step 1 holds, with a strength of 0.576. (See Figure 3-7.)

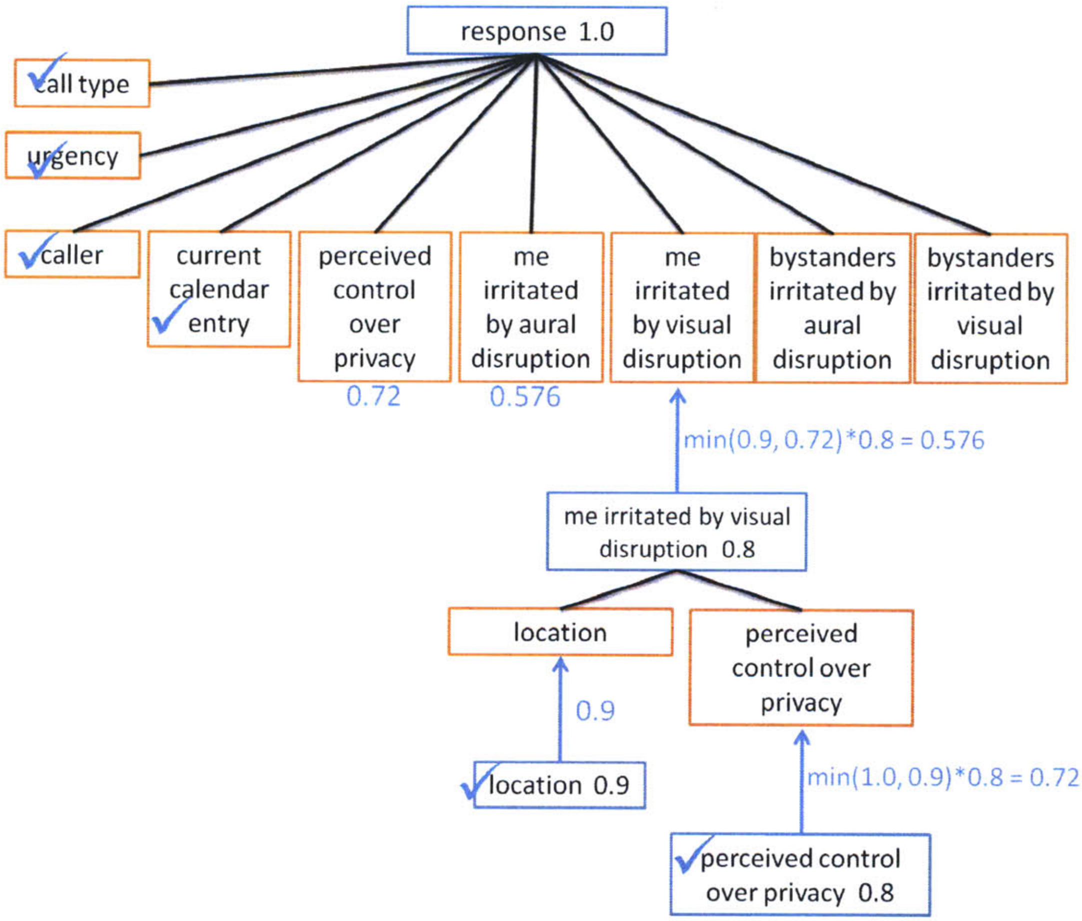

Figure 3-7: The rule in Step 6 is fired with strength of 0.576, which is the strength of the 7th condition of the first rule.

Step 7 – To figure out whether the 8^{th} condition of the rule in Step 1 holds, the system finds a rule:

```
=> bystanders_irritated_by_aural_disruption = true 0.7
perceived_control_over_privacy = low
location = performance_venue
```

which says that if the user seems to have low control over his privacy, and the user's location is a performance venue, then the system should believe with 0.7 strength that the bystanders (others around the user) will be irritated by aural disruption. This rule shares the same conditions as the previous rule, and hence the rule is fired with a strength of $min(0.72, 0.9) * 0.7 = 0.504$. And the 8^{th} condition of the rule in Step 1 holds, with a strength of 0.504. (See Figure 3-8.)

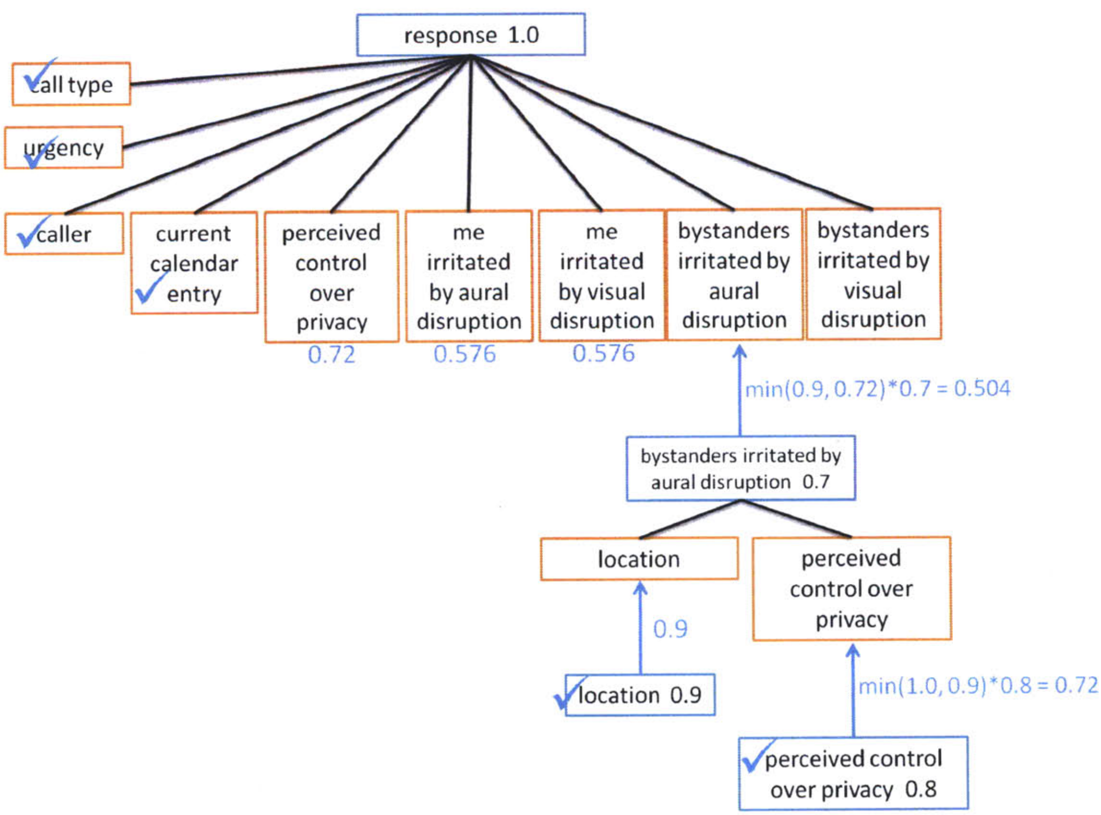

Figure 3-8: The rule in Step 7 is fired with strength of 0.504, which is the strength of the 8^{th} condition of the first rule.

Step 8 – To figure out whether the 9th condition of the rule in Step 1 holds, the system finds a rule:

```
=> bystanders_irritated_by_visual_disruption = true 0.7
perceived_control_over_privacy = low
location = performance_venue
```

which says that if the user seems to have low control over his privacy, and the user's location is a performance venue, then the system should believe with 0.7 strength that the bystanders (others around the user) will be irritated by visual disruption. This rule shares the same conditions as the previous rule, and hence the rule is fired, and the 9th condition of the rule in Step 1 holds, with a strength of 0.504. (See Figure 3-9.)

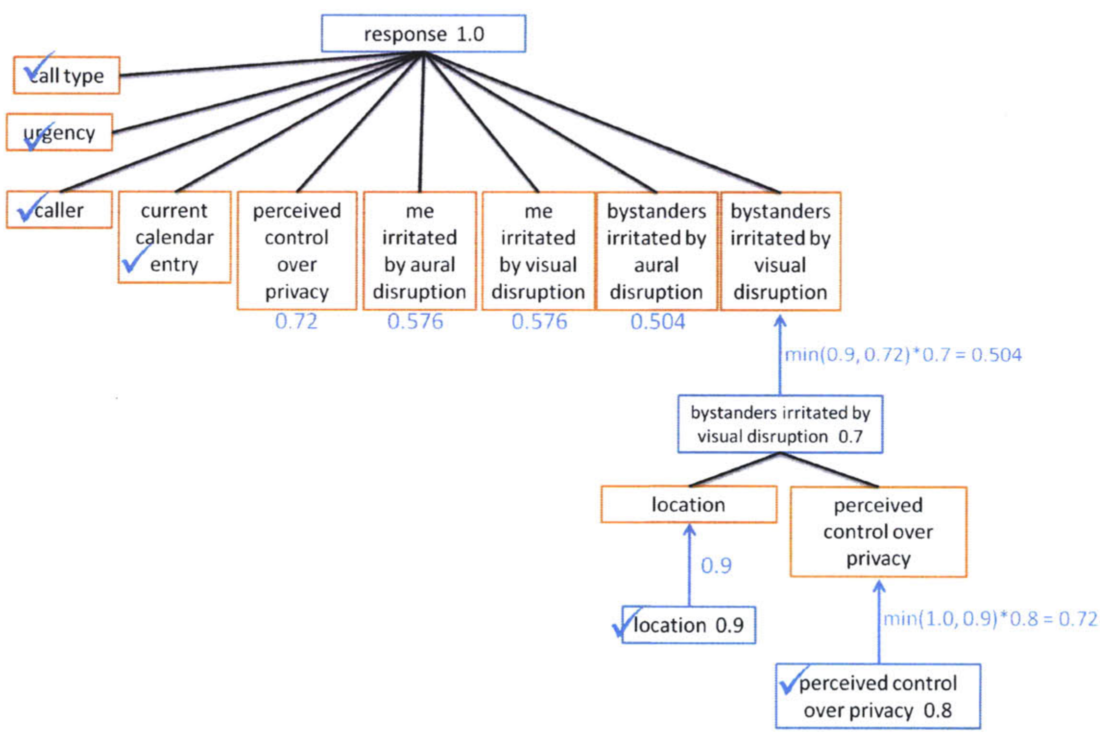

Figure 3-9: The rule in Step 8 is fired with strength of 0.504, which is the strength of the 9th condition of the first rule.

Step 9 – Looking at the first rule again:

```
=> response = (reveal-busy, vibrate) 1.0
call_type = voice
urgency != urgent
caller = coworker
current_calendar_entry = performing_arts
perceived_control_over_privacy = low
me_irritated_by_aural_disruption = true
me_irritated_by_visual_disruption = true
bystanders_irritated_by_aural_disruption = true
bystanders_irritated_by_visual_disruption = true
```

we now have the strength of belief of each condition in this rule, and the minimum is 0.504. Hence, the rule will fire with a strength of $0.504 * 1.0 = 0.504$. This is the rule that tells the device to respond with `reveal-busy` and `vibrate`. The decision-making system will explore each rule that fires (*i.e.* all of its conditions hold) and pick a conclusion that has the highest strength of belief as its final decision. (See Figure 3-10.)

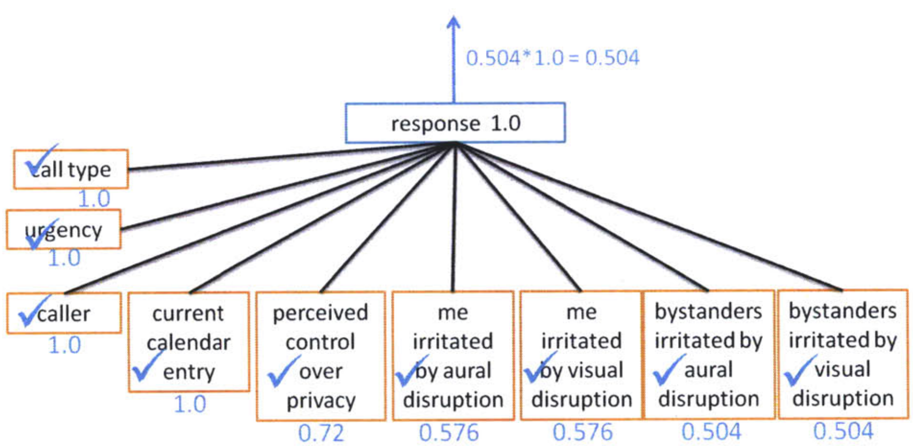

Figure 3-10: The weakest strength of belief in the conditions is 0.504; the rule is fired with a strength of $0.504 * 1.0 = 0.504$.

The concept of backward chaining and attaching a strength of belief to each piece of evidence is not new [8], and therefore it is not described in detail in this thesis.

The strengths of belief of the conclusion accumulate, if the conclusions are the same. In our system, this principle applies to all rules except the ones with the attribute `response`. To be more specific, if there are multiple rules that suggest the device to perform the same action (for example, multiple rules fire with the conclusion `=> response = (reveal-busy, vibrate)`), the strengths of these rules do not accumulate; the system simply picks the conclusion with the highest strength as its final recommendation to the device. One may argue that the principle should apply regardless of the attribute. However, in our rule set, rules with the same conclusion on `response` may share conditions: if the conditions in a rule are a subset of the conditions in another rule, the rule with fewer conditions is assigned with a lower strength of belief; the rule with more conditions is assigned with a higher strength of belief, because it requires more pieces of evidence to make all of its conditions hold. It would be unfair to accumulate the strengths of these particular rules because their conclusions in fact come from the same sources.

3.1.3 User Study: Vocabulary Collection

With the creation of rules, we came up with vocabulary that implicitly defined politeness for our system, with regard to handling voice calls and text messages. At the same time, we were curious about the vocabulary that is used by general users, and how much of the user vocabulary our vocabulary has captured.

Seven native speakers of English and eight non-native speakers were recruited for this study. The subjects were first given six scenarios (in class, at the movie, in a meeting, on a long bus trip, biking, and sleeping) where their phone would ring, and then they were asked to identify whether the phone was behaving (*i.e.* whether having the phone ringing was appropriate) in each scenario. Figure 3-11 shows the "votes" each scenario received for being identified as "phone misbehaving"; the X-axis indicates the number of subjects, and the Y-axis indicates the names of the scenarios.

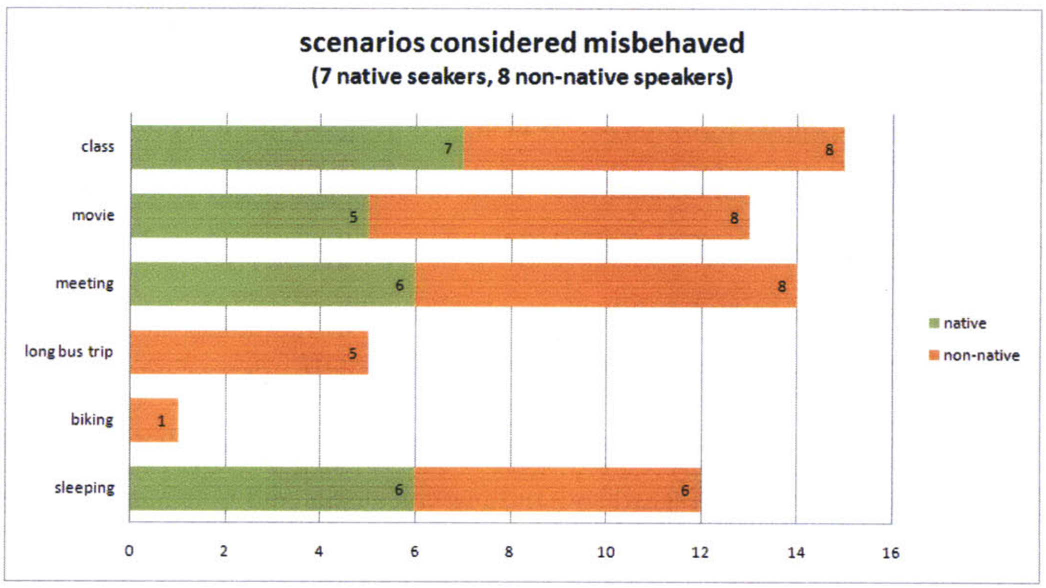

Figure 3-11: Number of subjects considering the phone misbehaving in each scenario.

Then, the subjects were asked why they would consider the phone misbehaving in the scenario(s) identified, and how they would teach the phone not to do this again if they were able to carry on a conversation with the phone. The verbal responses were transcribed to find out what vocabulary was used to describe the scenarios, and what vocabulary was used to instruct the phone what information to utilize.

Figures 3-12 to 3-17 show the vocabulary used by the subjects when describing why their phone would be considered misbehaving in each scenario. The transcriptions were processed by removing stop words and stemming each word. If a transitive verb is followed by different objects of different meanings (for example, "distract me", "distract the teacher"), the verb would be listed multiple times on the Y-axis based on the objects. The X-axis indicates the number of subjects who used the word/phrase; the color green represents subjects who are native speakers of English, and the color orange represents non-native speakers.

To compare the existing vocabulary in our system with the subjects' vocabulary, words in the existing vocabulary (attributes and values) were stemmed and individually compared with each word listed on the Y-axis. As long as the same word is found, or if the words are synonyms of each other (using WordNet [27]), we consider the word in our vocabulary has covered the subjects' intent. Words/phrases in the figures that are highlighted in yellow are the ones covered by existing vocabulary. The average vocabulary coverage for all six scenarios is 34%.

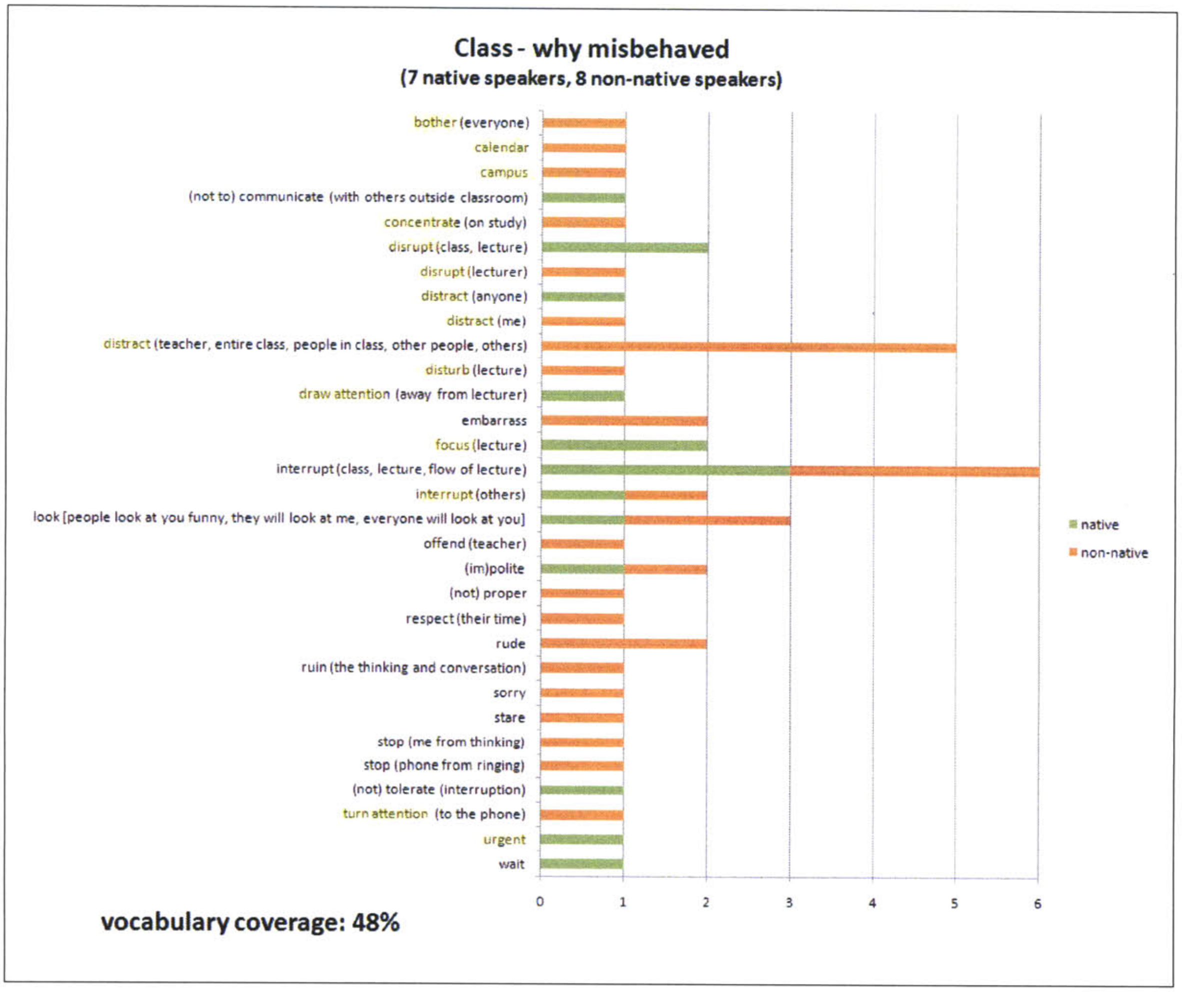

Figure 3-12: Vocabulary used by the subjects to describe why the phone misbehaved in the "class" scenario.

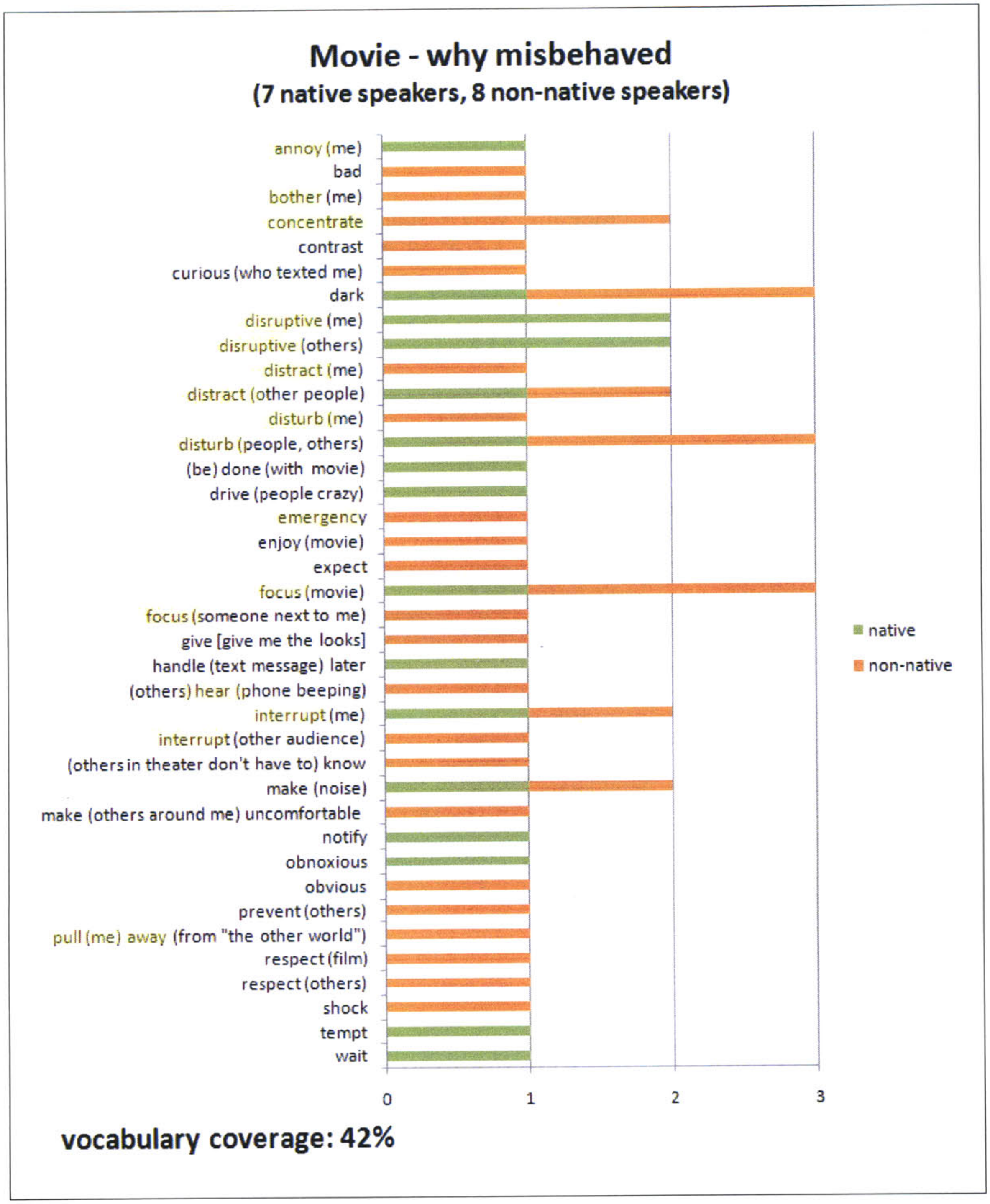

Figure 3-13: Vocabulary used by the subjects to describe why the phone misbehaved in the "movie" scenario.

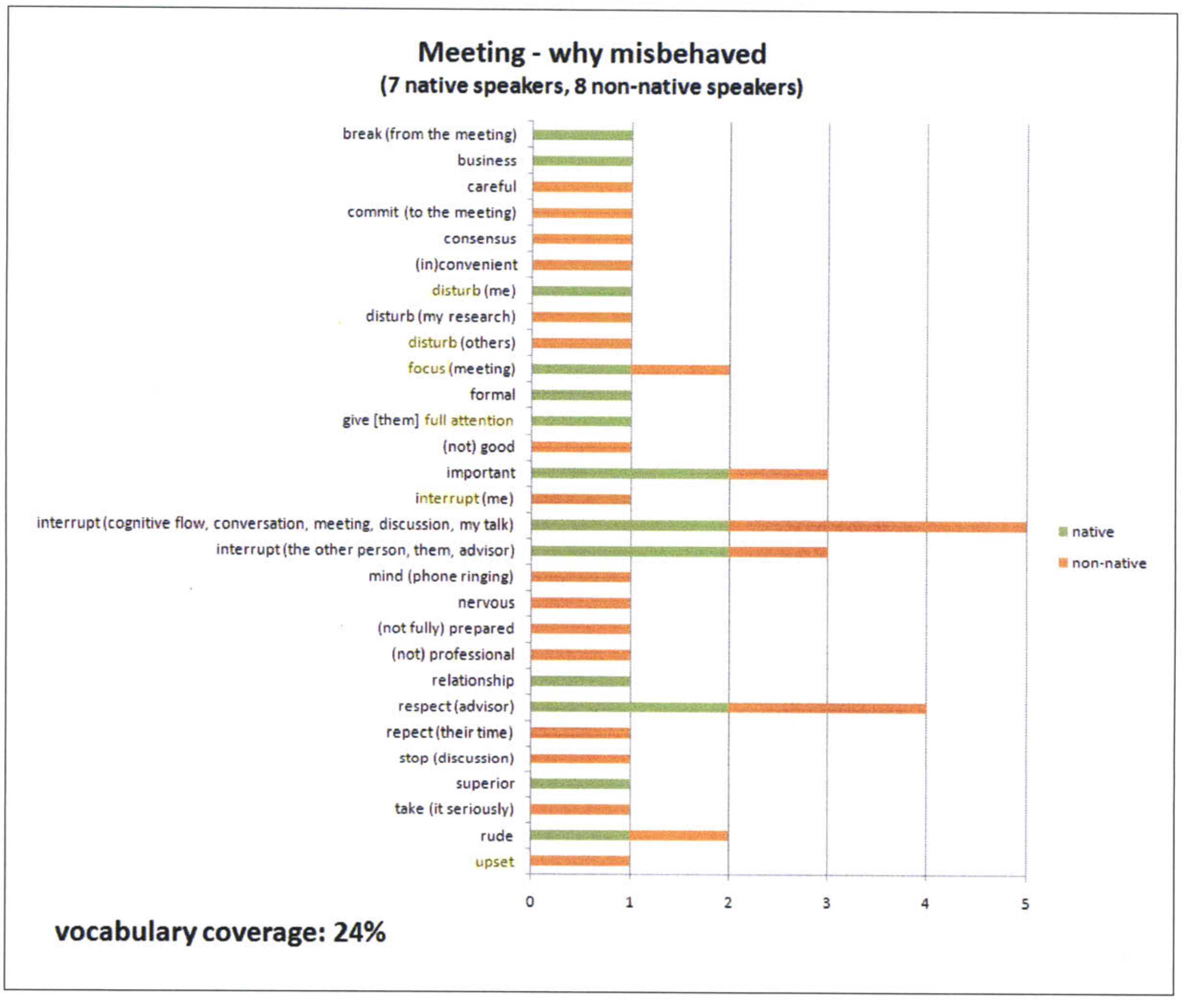

Figure 3-14: Vocabulary used by the subjects to describe why the phone misbehaved in the "meeting" scenario.

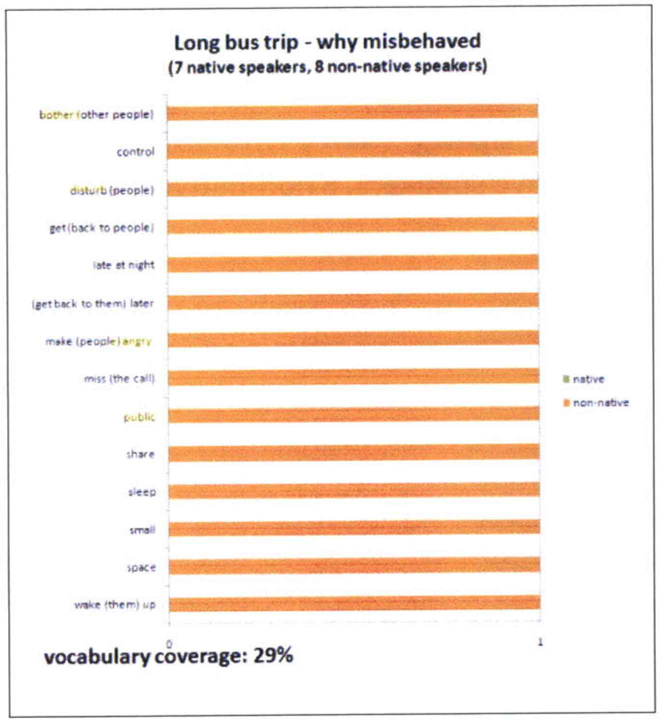

Figure 3-15: Vocabulary used by the subjects to describe why the phone misbehaved in the "long bus trip" scenario.

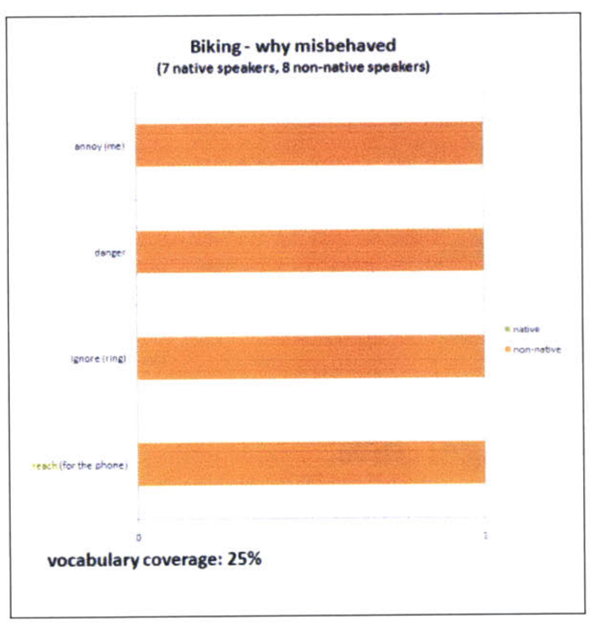

Figure 3-16: Vocabulary used by the subjects to describe why the phone misbehaved in the "biking" scenario.

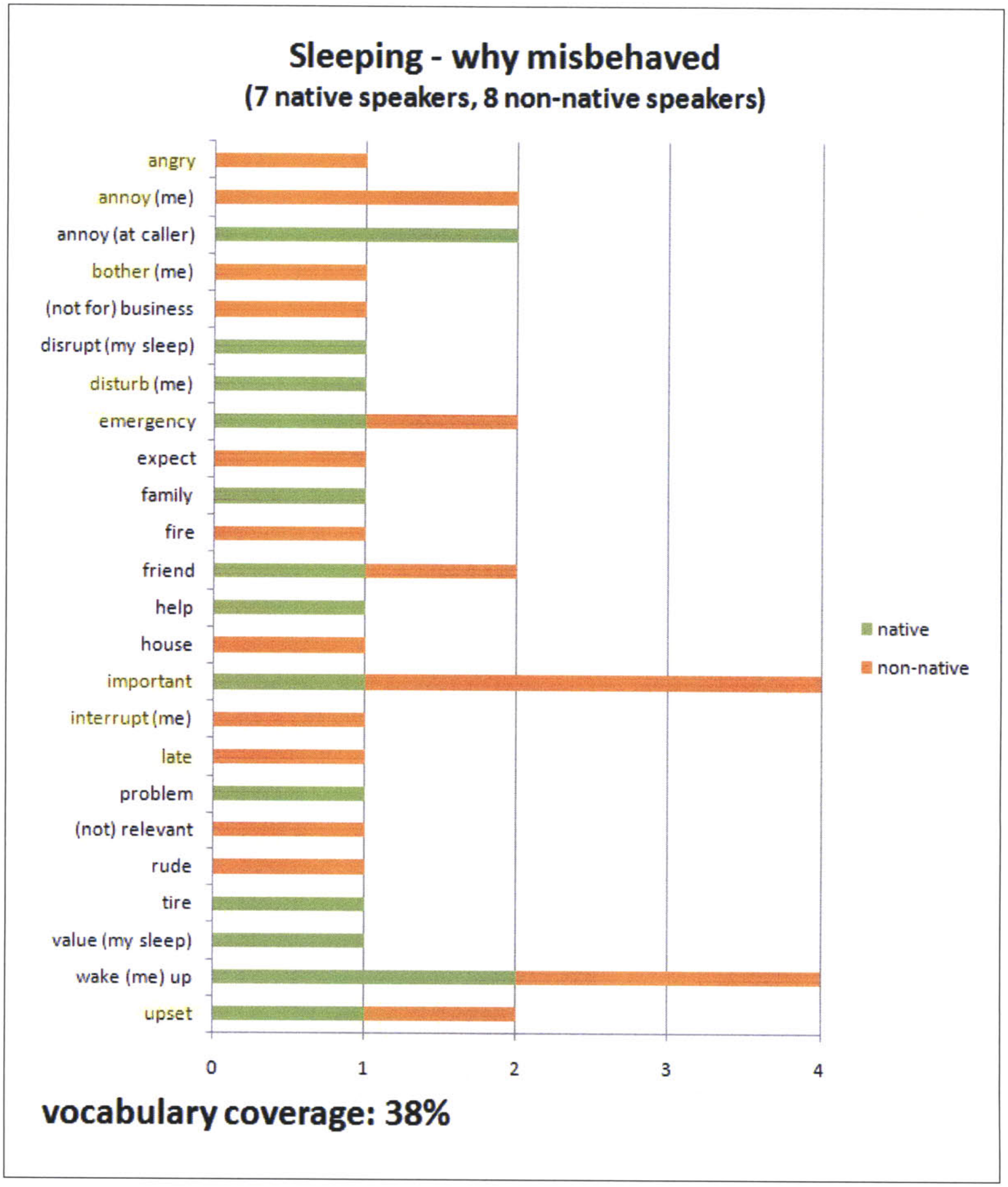

Figure 3-17: Vocabulary used by the subjects to describe why the phone misbehaved in the "sleeping" scenario.

Figures 3-18 to 3-22 show the vocabulary used by the subjects when describing how they would teach their phone to behave for each scenario. Using the same approach to compare the existing vocabulary in our system with the subjects' vocabulary, the average vocabulary coverage for all six scenarios is 42%.

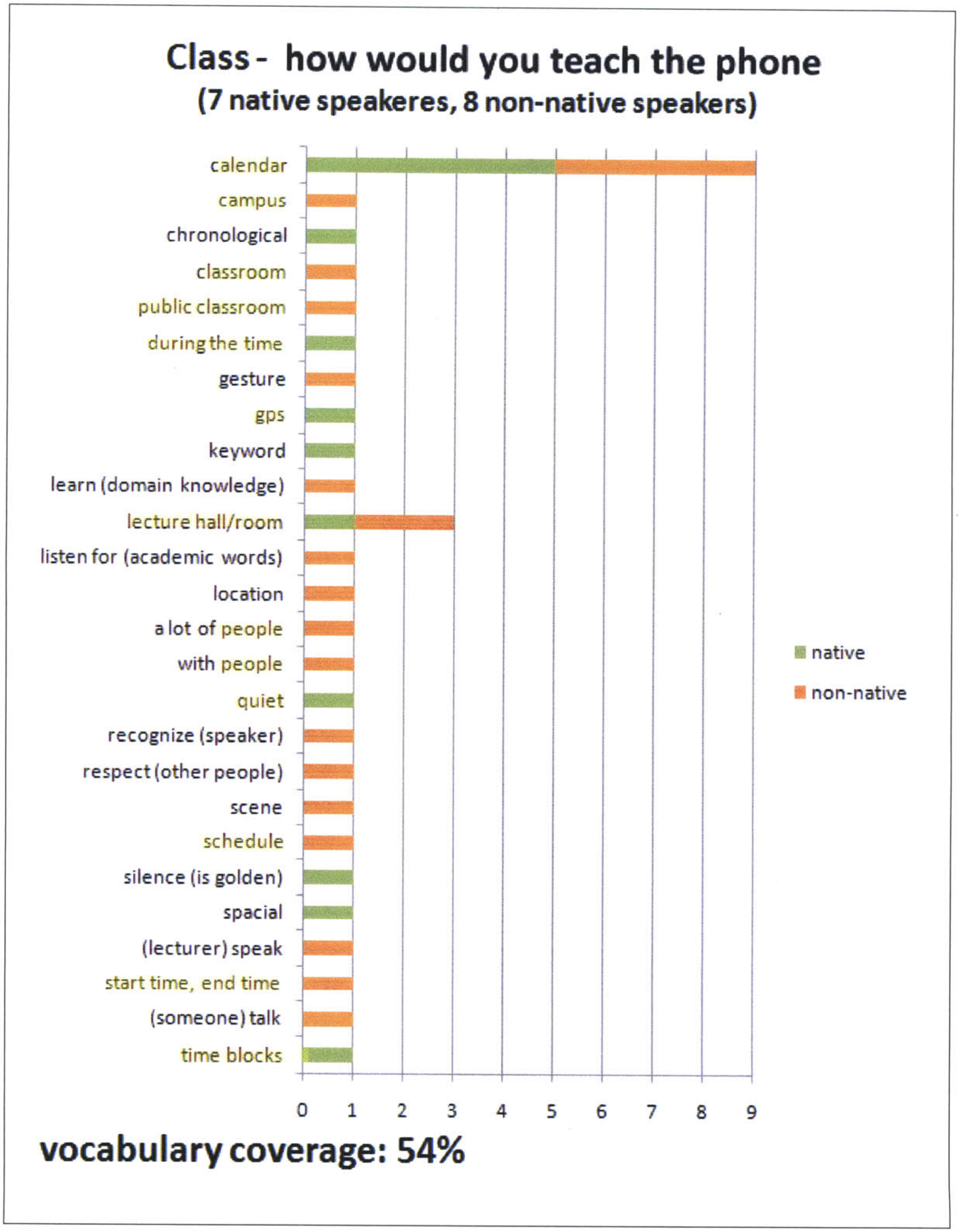

Figure 3-18: Vocabulary used by the subjects to describe how they would teach the phone to behave in the "class" scenario.

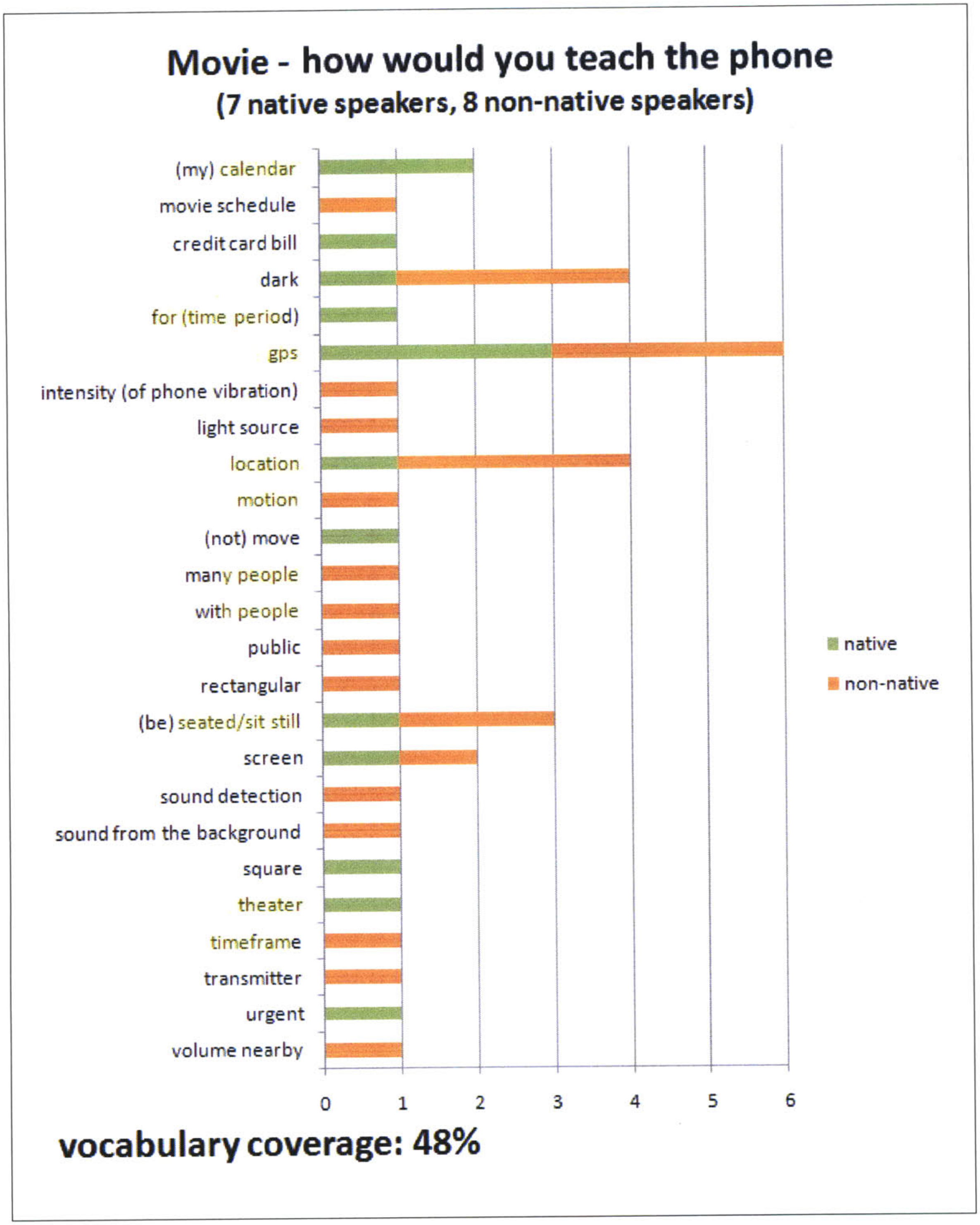

Figure 3-19: Vocabulary used by the subjects to describe how they would teach the phone to behave in the "movie" scenario.

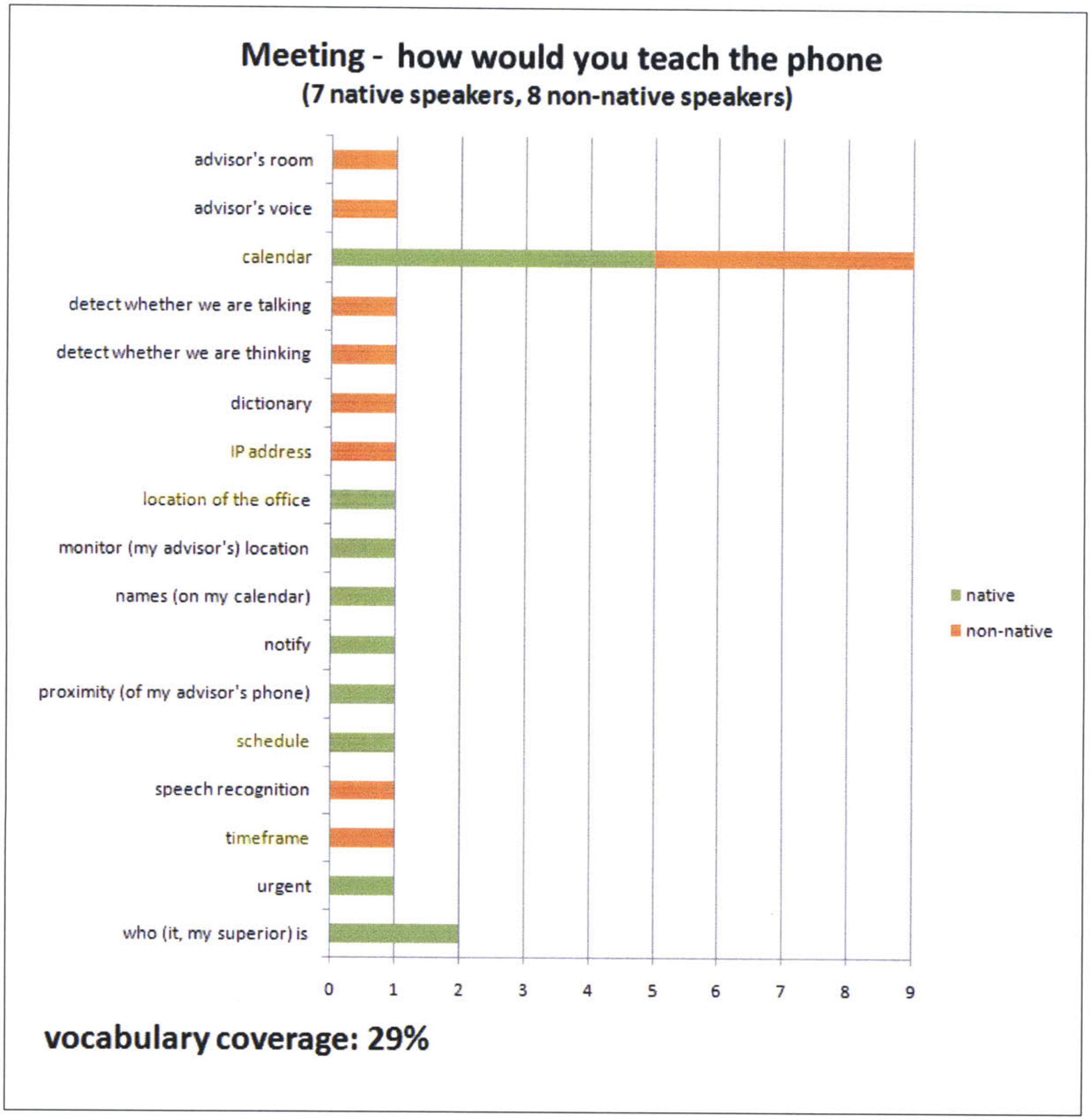

Figure 3-20: Vocabulary used by the subjects to describe how they would teach the phone to behave in the "meeting" scenario.

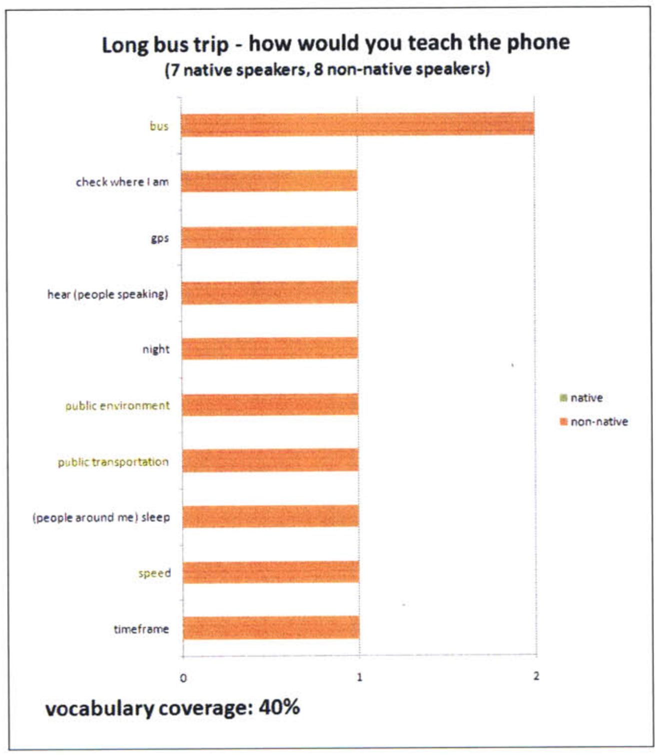

Figure 3-21: Vocabulary used by the subjects to describe how they would teach the phone to behave in the "long bus trip" scenario.

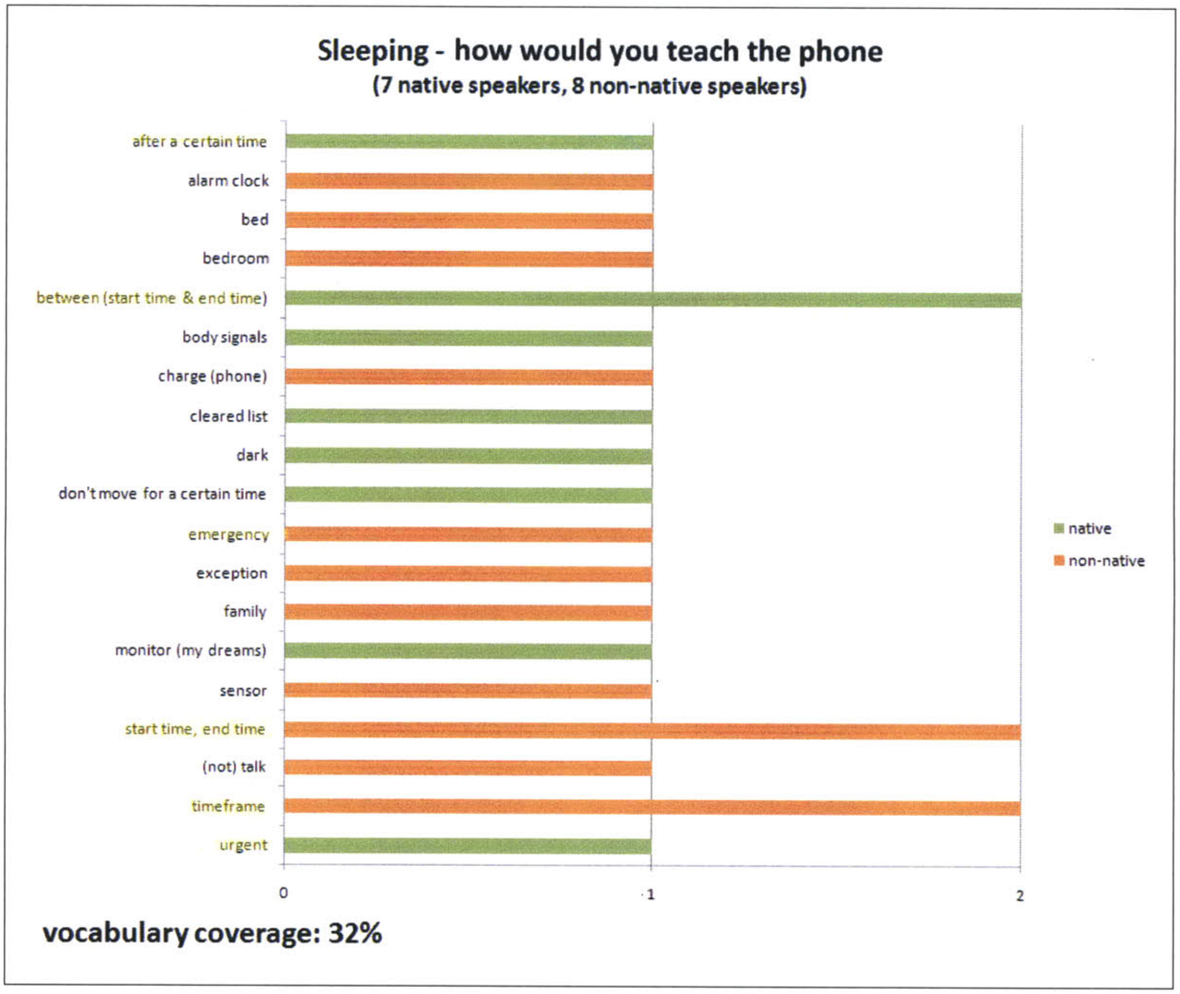

Figure 3-22: Vocabulary used by the subjects to describe how they would teach the phone to behave in the "sleeping" scenario.

If the vocabulary coverage were 100%, we could say our vocabulary has fully captured what general users use to define politeness for a mobile device. If the coverage were 0%, we could say we have a very different set of vocabulary than the general users when it comes to defining politeness for a device. With the current coverage, we see that this exploratory study shows us there are consequences of interruptions we did not think of (for example, the user gets stares from others if the mobile device rings in class), and there are technologies we might be able to utilize in the future when they become available (for example, the device should listen for "academic words" to find out that the user is in class).

3.1.4 User Study: Rule Understanding Study

The reasoning process of our system is transparent to the general users only when they are able to understand the rule set. To find out whether general users are able to understand the rules used in the system, we chose 25 rules from the existing rule set and asked the subjects to translate each rule to plain English, in the form of free text. The study was designed as a 5-day online game, and each day the subjects would be presented with 5 different rules. The subjects were recruited randomly online. Some subjects chose to participate in the game for multiple days, while others only participated in one of the 5-day game. Subjects were asked to identify themselves as programmers or non-programmers, because we assume that programmers would be more comfortable interpreting expressions written in the form of "attribute, operator, value".

Figure 3-23 is the breakdown of the subject pool: There were 32 unique participants in total, with 10 programmers and 2 non-programmers as returning participants. Every day there were more programmers than non-programmers, and hence for the analysis of the result, we focus on the responses from programmers.

The following text would be presented to the subject at the beginning of the game each day:

"*You are hired by a high-level executive, who has a set of rules (written in an unusual form of language) on how his incoming phone (voice) call and text messages* *should be handled."*

"Translate the rules into plain English, so that the assistant who handles the phone calls and messages can easily learn from you what the executive wants."

Then the subject would see 5 rules to be translated. The "unusual form of language" is in fact the syntax of the rule, in the form of "attribute, operator, value". An example rule is shown below. Notice that the syntax here is different from the syntax shown in section 3.1.2. The `IF` and `THEN` are now mentioned explicitly, instead of being represented by a symbol (or no symbol). The conditions are mentioned prior to mentioning the conclusion. The purpose of changing the syntax in this study is to conform to the "If ... then ..." sentence structure in English, and to reduce the subjects' cognitive burden of figuring out the syntax – we were interested in whether the subjects could understand the meaning of a rule, which should be independent of the syntax.

```
IF
call_type = text_message
caller = important_caller
others_willingness_redirect_visual = low
my_willingness_redirect_visual = low
THEN
response = vibrate, certainty = 0.8
```

The rule above means *"If I receive a text message from someone important, others are not willing to redirect their visual attention, and I am not willing to redirect my visual attention, then the phone should be 80% certain that it should vibrate."*

When comparing a subject's interpretation of the rule against the meaning of the rule, we would like to know much overlap in vocabulary there is between the two. Suppose a subject who did not completely understand the rule came up with such an interpretation: *"If it's a text from someone important, and there seem to be a lot of other people around, then vibrate my phone"*, when we look for vocabulary matches, we look at the verbs, adjectives, adverbs (indicating strength of certainty) in their stemmed form; we also look at their synonyms with the help of WordNet [27], as well as phrases that convey the same meaning as these words. The stopwords are ignored, but words indicating negation are kept in the interpretation as they are crucial to correctly interpreting the meaning of the rule.

The nouns are manually compared with different levels of strictness. Due to the design of the study, the noun (noun phrase) indicating the receiver of a call in the "if" clause could be "the executive", "I", "your boss", "you", "the phone", or "it"; and the noun (noun phrase) indicating the agent of responding the call in the "then" clause could be "the assistant", "you" (sometimes omitted when the sentence is imperative), "the phone", or "I". These terms are generally not considered in the matching process because they were only used to indicate the receiver/agent roles. Meanwhile, nouns (noun phrases) indicating the caller, the user, and the bystanders are considered in the matches, because they are crucial to understanding the conditions of how a call should be handled.

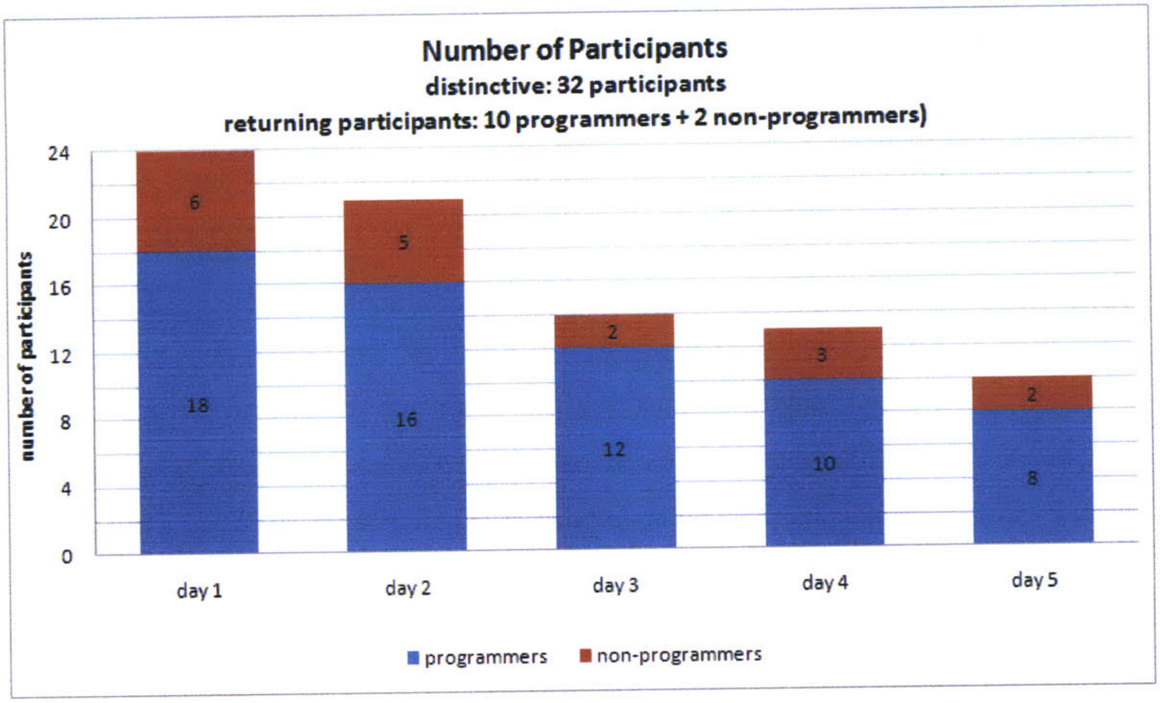

Figure 3-23: Number of participants of the rule understanding study

Subject's interpretation:

If it's a text from someone important, and there seem to be a lot of other people around, then vibrate my phone

The rule means:

If I receive a text message from someone important, others are not willing to redirect their visual attention, and I am not willing to redirect my visual attention, then 80% of the time the phone should vibrate.

Figure 3-24: Matching subject's interpretation against the meaning of the rule. Text highlighted in blue indicates a match; text highlighted in red indicates a mismatch.

As demonstrated in Figure 3-24, text highlighted in blue indicates a match; text highlighted in red indicates a mismatch. To be more specific, terms that should be matched in the meaning are:

1. text message
2. someone important
3. others
4. not willing (tied to "others")
5. redirect (tied to "others")
6. visual attention (tied to "others")
7. I
8. not willing (tied to "I")
9. redirect (tied to "I")
10. visual attention (tied to "I")
11. 80% certain
12. vibrate

The subject's interpretation of the rule includes 4 (out of 12) matches; in other words, there is a 33% match. Hence, we can say that the subject has 33% understanding of the rule. Yet, we can be harsh in the evaluation and look at whether or not the subject's interpretation is a perfect match of the meaning of the rule, because in real life, if the assistant (or the device) misunderstands the rule, it is very likely that a wrong decision will be make when a call is received.

By looking at the subjects' interpretations in a binary sense (*i.e. whether or not a subject understood a rule perfectly*), we have a visual representation, shown in Figure 3-25, of the percentage of subjects having perfect understanding of each rule. Data points in red are collected from non-programmers; data points in blue are collected from programmers. The X-axis contains the label for the rules: for example, 2-1 means the first rule a subject would see on day 2 of the study. Because each day the number of participants were different, the Y-axis is labeled with the percentage of subjects (programmers and non-programmers are accounted for separately). Since programmers made up the majority of the subject pool, the data points are sorted based on the percentage of programmers that understood a rule perfectly. From the figure, we can see that 2/3 of the rules were understood perfectly by at least 50% of the programmers.

In the study, subjects were asked to type "I don't understand" if they could not understand a rule. Some subjects would insert question marks in their interpretations to indicate that they did not understand part of a rule. To find out how many subjects were unable to understand a rule at all, we have a visual representation, shown in Figure 3-26, of the percentage of subjects unable to understand a rule at all. The X- and Y-axes are labeled the same way as the previous figure. For the ease of visualization, the data points are sorted based on the percentage of non-programmers that were unable to understand a rule at all. From the figure, we can see that in the worst case, only 25% of the programmers were not able to understand a rule.

Based on the subjects' interpretation, we have found that there are several terms in the vocabulary that were considered confusing or difficult to understand:

1. "others" could be interpreted as the caller.

2. "redirect" could be interpreted as redirecting a voice call to text message or voicemail, the likelihood of being eavesdropped, or the priority of the incoming information.

3. "perceived privacy" could be interpreted as the level of confidentiality of the incoming information, or the user-desired privacy.

4. the meaning of "acceleration" and "orientation" were considered unclear.

Due to these confusions, some of the attributes in the vocabulary are reworded:

- `others_willingness_redirect_hearing` is reworded as `bystanders_irritated_by_aural_disruption`
- `others_willingness_redirect_visual` is reworded as `bystanders_irritated_by_visual_disruption`
- `my_willingness_redirect_hearing` is reworded as `me_irritated_by_aural_disruption`
- `my_willingness_redirect_visual` is reworded as `me_irritated_by_visual_disruption`
- `perceived_privacy` is reworded as `perceived_control_over_privacy`
- `acceleration_orientation_pattern` is reworded as `moving_pattern`

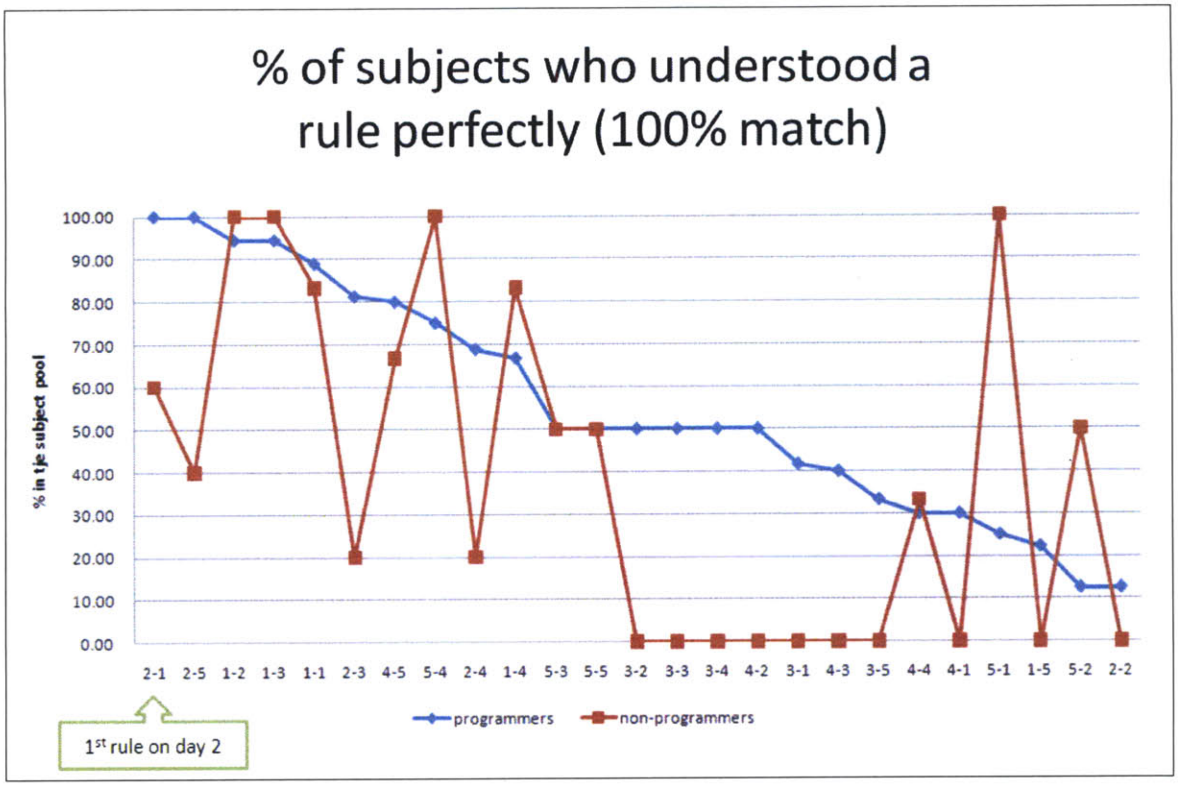

Figure 3-25: Percentage of subjects who understood a rule perfectly

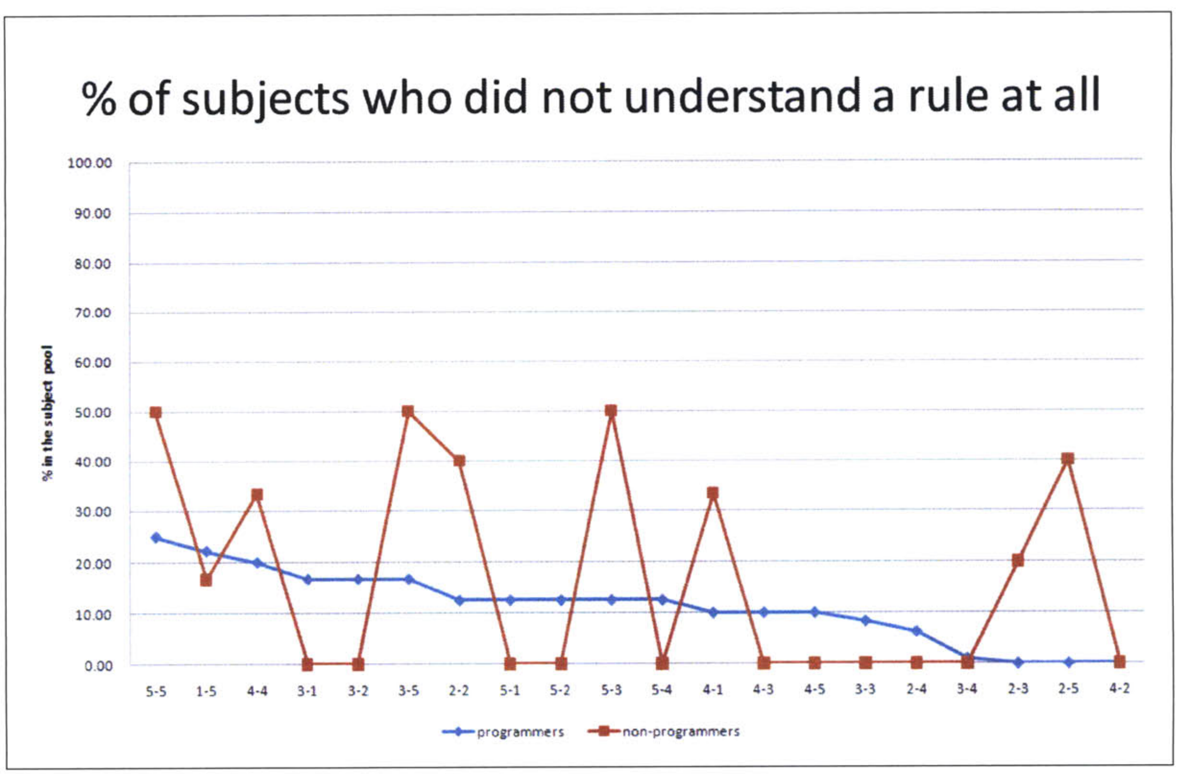

Figure 3-26: Percentage of subjects unable to understand a rule at all

3.1.5 User Study: Rule Writing Study

In order for a decision-making system to be customizable, users should be allowed to modify existing rules and create new rules, so that the behavior of a device can be polite in its owner's point of view. Before we went ahead to design an interface for such customization, we first wanted to know whether general users are able to write rules, using the vocabulary defined in our work, to describe scenarios they expect the device to behave politely.

The same 15 subjects (9 programmers and 6 non-programmers) in the vocabulary collection study were asked to perform tasks in this study. Among the 9 programmers, 4 had participated in a previous study (the rule understanding study or the calendar data collection study). All 15 subjects were introduced to the vocabulary of our system (Tables 3.1 to 3.4), the "`IF` ... `THEN` ..." syntax for writing the rules, and the concept of backward chaining, which is typically used in rule-based systems. After the introduction, the subjects were asked to write rules describing the scenarios previously identified as "phone misbehaving", using the existing vocabulary in our system. Subjects were free to write their rules on a piece of paper, or to type the rules using a computer with a plain text editor.

To define whether a subject has successfully written a rule, the criterion we use is whether the subject has written a rule to convey what was previously said out loud. For example, a subject who did not want his phone to ring in the "in class" scenario would say, "*I'd tell my phone if you see the word 'class' on my calendar, and my location is the same as what I wrote on my calendar, then you know I'm in class.*" Then the subject would write down

```
IF
current_calendar_entry = class
location = calendar.current.location
THEN
scenario = class
```

The attributes used by the subject in this rule were all defined in our vocabulary.

Although the value `calendar.current.location` requires that the system perform beyond its current capability, we still consider that the subject has successfully written a rule. Even if a subject's verbal expression does not seem to make common sense (for example, "*if my moving speed is 30 miles per hour, then I'm in class*"), as long as the subject was able to come up with a rule to describe what had previously been verbalized (`IF moving_speed = 30 THEN scenario = class`), we consider this a success.

There are multiple ways to fail this rule-writing test: if a subject came up with a rule that did not describe what had been verbalized, it is considered a failure; if a subject was unable to finish the task (writing rules for at most 6 scenarios in 30 minutes) in time, it is considered a failure; if a subject had trouble writing down what had been verbalized, it is also considered a failure.

Among all 15 subjects, 1 programmer could not finish the task in time, and 1 non-programmer had trouble writing down what she had verbalized.[1] Out of 64 attempts, 57 were successful (89%). We can say that most subjects were able to write rules successfully for their own needs. However, we were curious to what extent has the existing rule set in our system captured the subjects' view of "the world" – in this study, the six example scenarios.

Observing the rules written by the subjects, we find that most of the rules only made use of the raw signals available to the device, and almost no backward chaining was used to construct inferred information that might be useful for the device to figure out the scenario. If we simply compare the subject's rule:

```
IF
current_calendar_entry = class
location = calendar.current.location
THEN
scenario = class
```

[1]The subject said, "*I want my phone to check my calendar. If my calendar says 'class', then you should vibrate.*" When being asked to write this down as rule for her phone, the subject responded, "*I don't know what you want me to write; the phone should automatically check my calendar.* ".

with the existing rule in our system that describes the same scenario:

```
IF
perceived_control_over_privacy = low
bystanders_irritated_by_visual_disruption = true
bystanders_irritated_by_aural_disruption = true
me_irritated_by_visual_disruption = true
me_irritated_by_aural_disruption = true
THEN
scenario = class
```

we can easily say that there is no vocabulary overlap between these rules. However, once we expand the existing rule with lower-level rules that make use of raw signals available to the device, we will end up with a rule that is more comparable to the subject's rule. To demonstrate this in detail, the 5 conditions in the existing rule can each be expanded separately by another rule existing in our system:

```
IF
location = public_location
moving_speed_mph < 3
number_of_companions > 20
THEN
perceived_control_over_privacy = low

IF
current_calendar_entry = class
location = public_location
THEN
bystanders_irritated_by_visual_disruption = true

IF
current_calendar_entry = class
```

```
location = public_location
THEN
bystanders_irritated_by_aural_disruption = true

IF
current_calendar_entry = class
location = public_location
THEN
me_irritated_by_visual_disruption = true

IF
current_calendar_entry = class
location = public_location
THEN
me_irritated_by_aural_disruption = true
```

When we replace the 5 conditions with the conditions used in the 5 rules above, we have a new rule:

```
IF
current_calendar_entry = class
location = public_location
moving_speed_mph < 3
number_of_companions > 20
THEN
scenario = class
```

which is more comparable to the rule written by the subject.

We take into account the fact that most of our subjects did not have prior exposure to the vocabulary defined in our system, and that the subjects might not be aware of all the possible attributes that could be used to describe a scenario. Therefore, when comparing the existing rule with a subject's rule for vocabulary overlap, we consider how much of the subject's rule has the existing rule covered. In the example we have brought up here (Figure 3-27), the first condition in both rules are exactly the same. The second condition in the subject's rule shares the same attribute (`location`) as the existing rule, but the values are different; we consider this condition as a mismatch. There are 2 conditions in the subject's rule, and the exiting rule covers 1 condition; therefore there is 50% of vocabulary overlap.

Subject's rule

```
IF
curren_calendar_entry = class
location = calendar.current.location
THEN
scenario = class
```

Existing rule, expanded

```
IF
current_calendar_entry = class
location = public_location
moving_speed_mph < 3
number_of_companions > 20
THEN
scenario = class
```

Figure 3-27: Subject's rule and the existing rule for the "in class" scenario. The condition in blue indicates a match in vocabulary, while the condition in red indicates a mismatch.

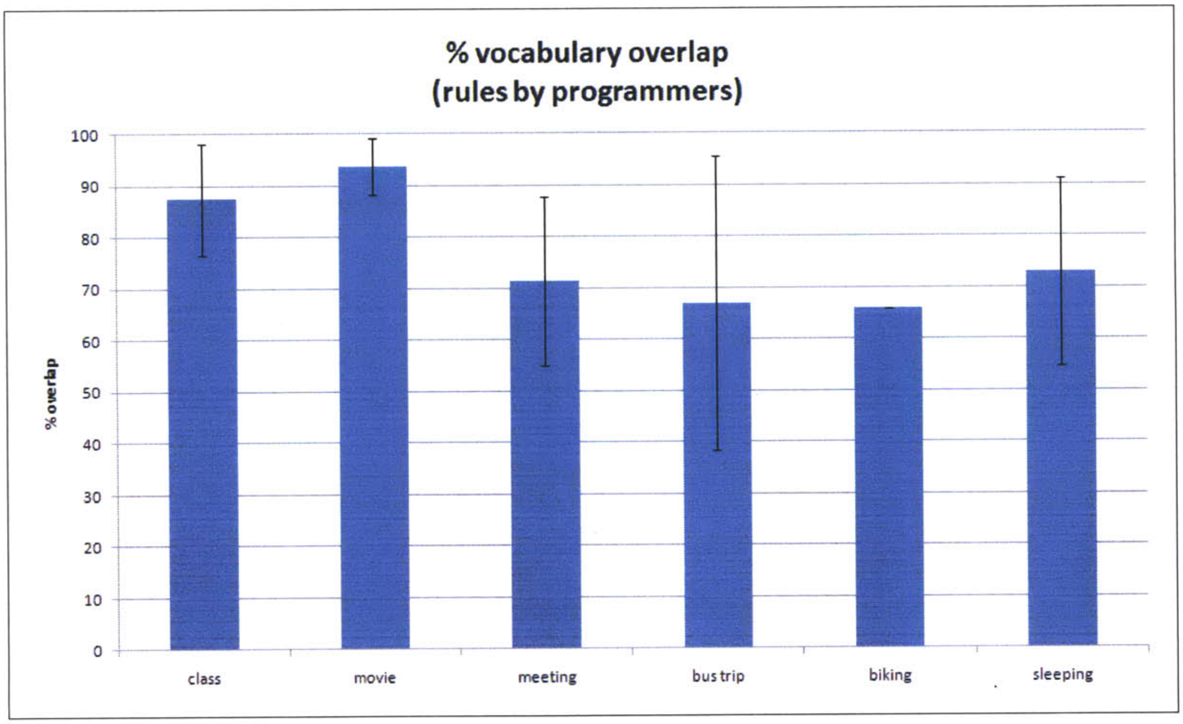

Figure 3-28: Averaged percentage of vocabulary overlap for rules produced by programmers.

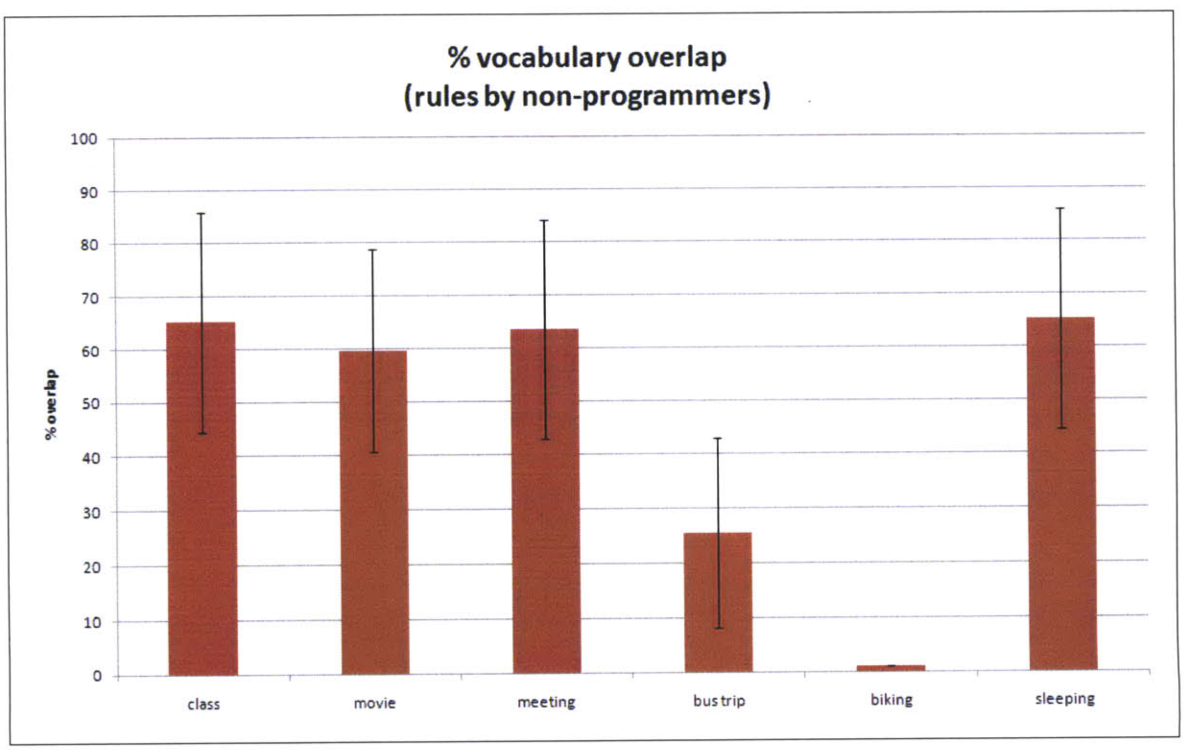

Figure 3-29: Averaged percentage of vocabulary overlap for rules produced by non-programmers.

Applying the same approach to all the rules produced by our subjects, we have the percentage of vocabulary overlap between the existing rule set and the subjects' rules for each scenario. Figures 3-28 and 3-29 shows the averaged percentage of vocabulary overlap for rules produced by programmers and non-programmers, respectively. The X-axis contains all 6 scenarios in the study. The Y-axis indicates the averaged percentage overlap; that is, how much has the existing rule set covered the rules produced by the subjects. For rules produced by programmers, at least 50% of the vocabulary is covered by the existing rule set in most cases; for rules produced by non-programmers, at least 40% of the vocabulary is covered by the existing rules set in most cases.

Given the percentage of vocabulary overlap, it is difficult to conclude how well or how poorly our vocabulary has covered the general users' vocabulary. However, this exploratory study has shown us what other capabilities the users expected our system to possess: for example, comparing tags on the map with calendar entries, creating new attributes and using them as flags to keep track of chronological changes of user location, etc.

3.2 Building a Knowledge Base

Even though the current system in our work has 250 rules to cover scenarios where a device is expected to behave politely, the rules are certainly not comprehensive enough to cover all the possible scenarios in life. Besides, different users may have different expectations for the device given the same scenario. Based on these concerns, we constructed an interface to allow the users to debug the knowledge base when the device misbehaves, and to customize the device or to build knowledge of the world by creating new rules.

To be more concrete, a device equipped with our decision-making system is likely to misbehave when any of the following happens:

1. there is no rule that covers a specific conclusion (a new rule needs to be created)

2. a rule is too specific (one or more condition needs to be removed)

3. a rule is too general (one or more condition is missing)

4. the threshold (for example, the maximum/minimum moving speed) is set incorrectly

5. the certainty factor (strength of belief) in a conclusion is set too high or too low

In the next section, we demonstrate an example where a rule is too general, and show how a user could fix this by adding a condition using the interface we propose.

3.2.1 Example Scenario

Suppose a user received a call from his labmate while in a meeting with a professor, and the device rang. The user is curious what rule(s) caused the device to ring, and wants to fix the rule(s) by using the debugging interface (as illustrated in Figures 3-30 to 3-44):

1. The interface displays a list of call and message histories, as a regular cell phone would do. The system also displays the response of the device when each incoming call/message took place, to allow the user to identify the instance of misbehavior.

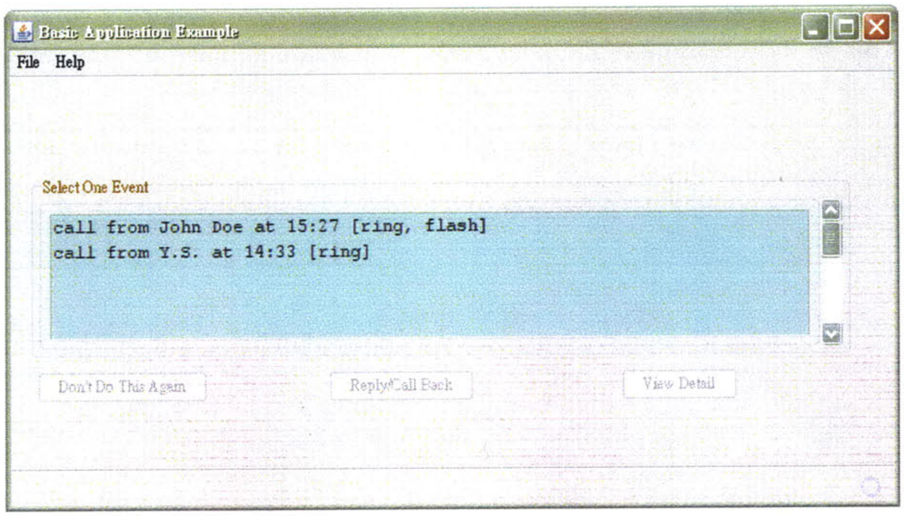

Figure 3-30: The interface displays a list of call and message histories.

2. The user selects an instance where the device was not expected to ring. The "Don't Do This Again" button allows the user to view the facts known to the device when the misbehavior took place.

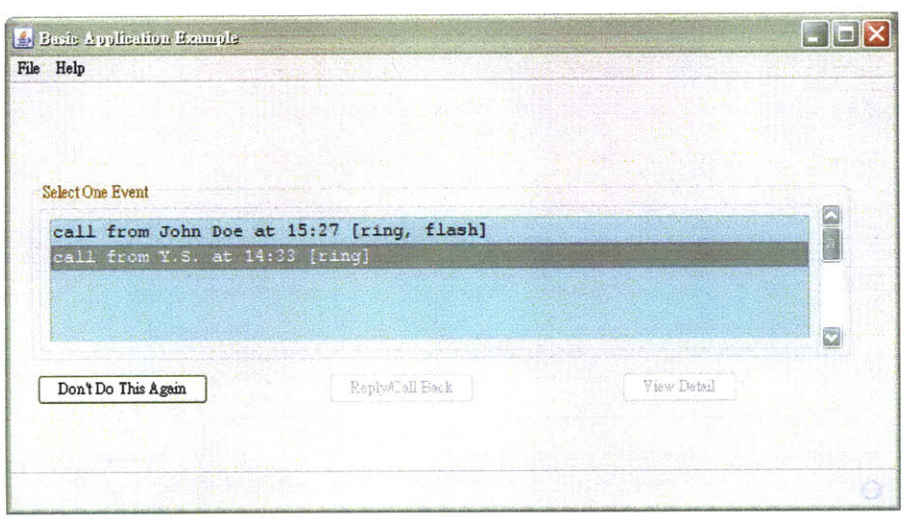

Figure 3-31: The user selects an instance where the device was not expected to ring.

3. The interface displays the facts available to the device when the instance took place. In the future, the interface may be augmented to allow the user to explore the rules that directly make use of a particular fact in the condition.

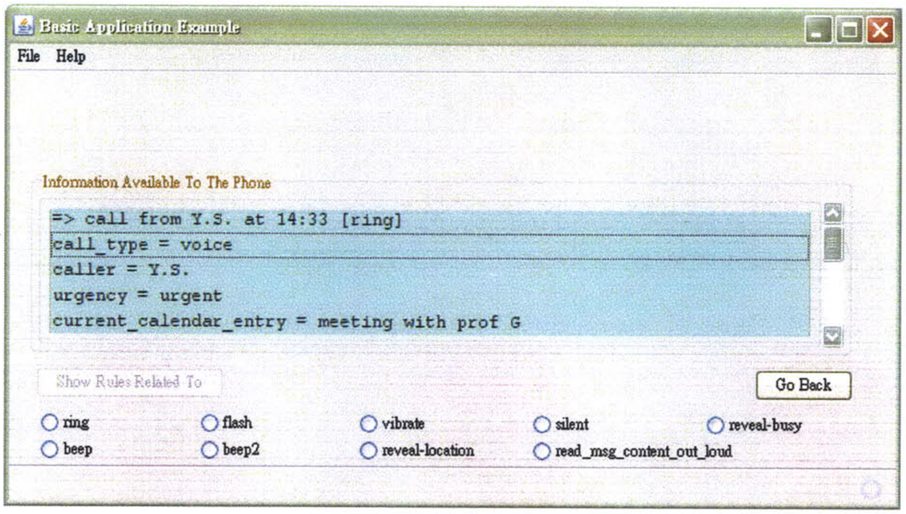

Figure 3-32: The interface displays the information available to the device when the misbehavior took place.

4. The user wants to see rules that might have caused the device to ring. The user can also choose to see why the device did not flash, vibrate, etc. by selecting another behavior.

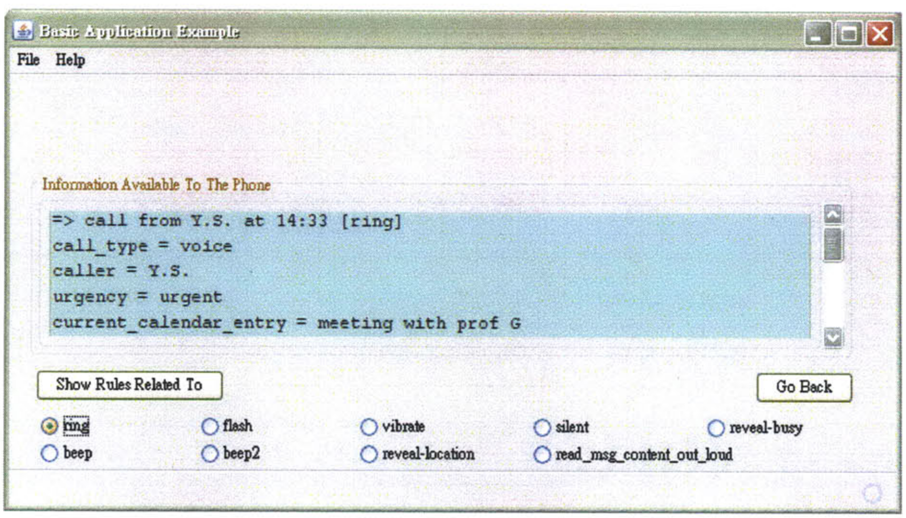

Figure 3-33: The user wants to see rules that might have caused the device to ring.

5. The interface displays the rule numbers of rules related to ringing. The visual display of the rule numbers remains flat, no matter where in the tree structure a rule is (see Figures 3-1 to 3-10 for tree structure examples).

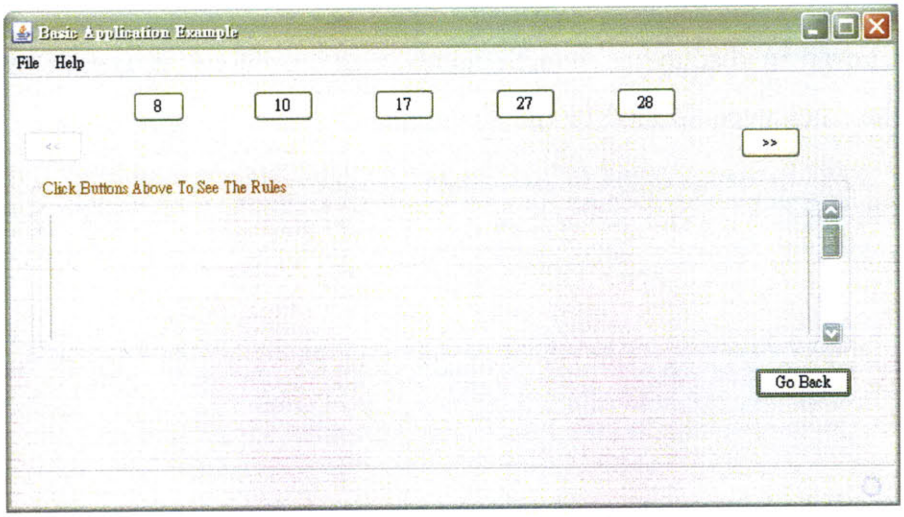

Figure 3-34: The interface displays the rule numbers of rules related to ringing.

6. The user is able to view the content of the rules by clicking on the rule numbers. The user clicks on rule #8, and the interface informs the user that the rule selected was not applied to the situation.

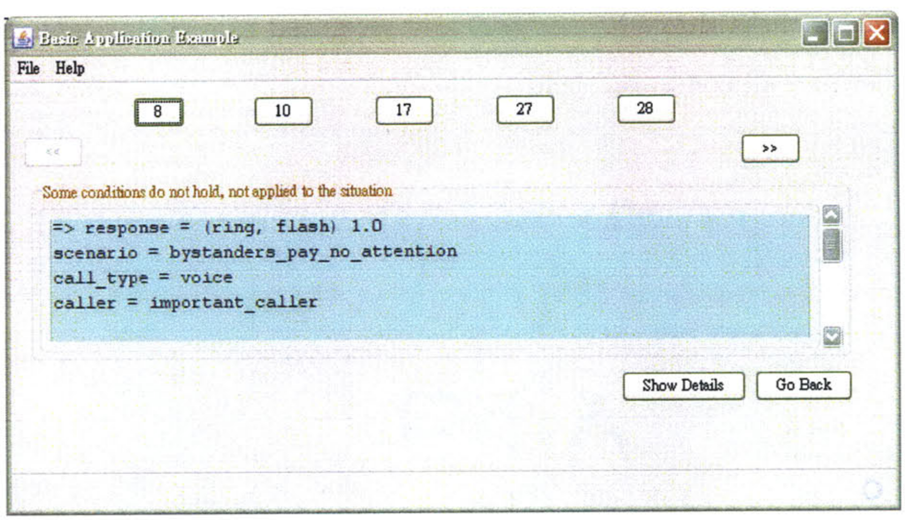

Figure 3-35: The user is able to view the content of the rules by clicking on the rule numbers.

7. The user clicks through rules #10 and #17. Neither rule was applied to the situation, because some conditions did not hold. The user keeps clicking through rules #27 and #28, but neither was applied to the situation. Then the user proceeds to explore more rules by clicking on the >> button. In the future, the interface can be improved by displaying the rule numbers in a different manner, depending on whether the rules were applied to the situation.

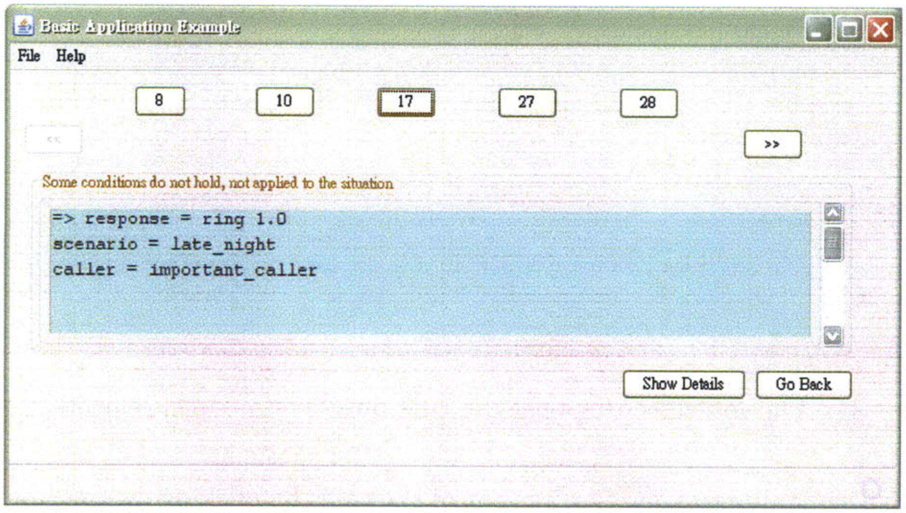

Figure 3-36: The rule selected (#17) was not applied to the situation, because some conditions did not hold.

8. The user clicks through rules #29 and #74, and rule #74 was applied to the situation because all conditions held.

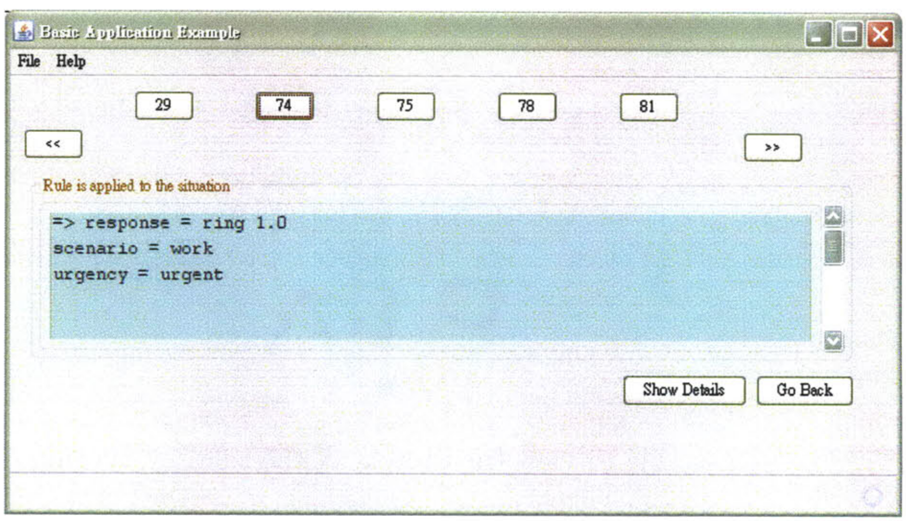

Figure 3-37: The rule selected (#74) was applied to the situation.

9. The user sees that the system thought the scenario was "work" when the misbehavior took place (the urgency information is a known fact). The user then selects the scenario condition and clicks on the "Show Selected" button to find out which rule caused the condition to hold.

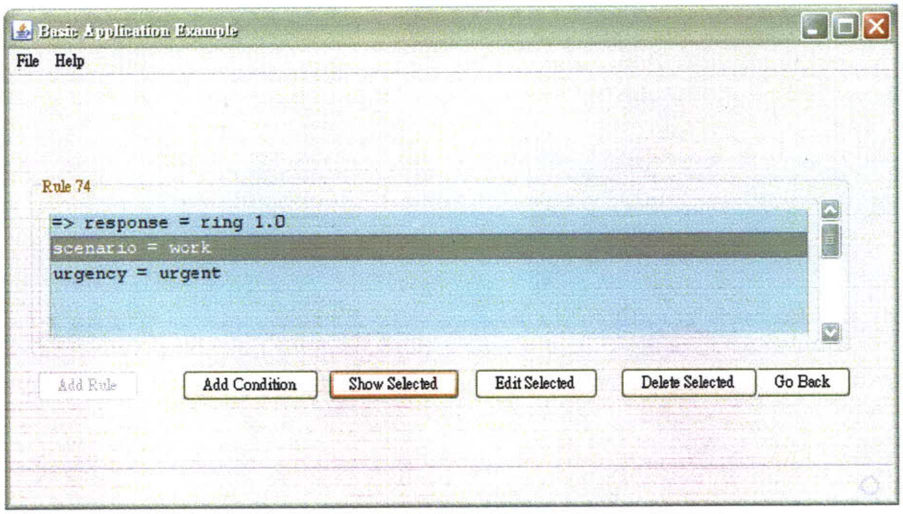

Figure 3-38: The user explores further to see why the system thought the scenario was "work".

10. Rule #73 caused the scenario condition to hold, because the time of day was between 9am and 6pm. The user thinks that the rule makes sense but is too general.

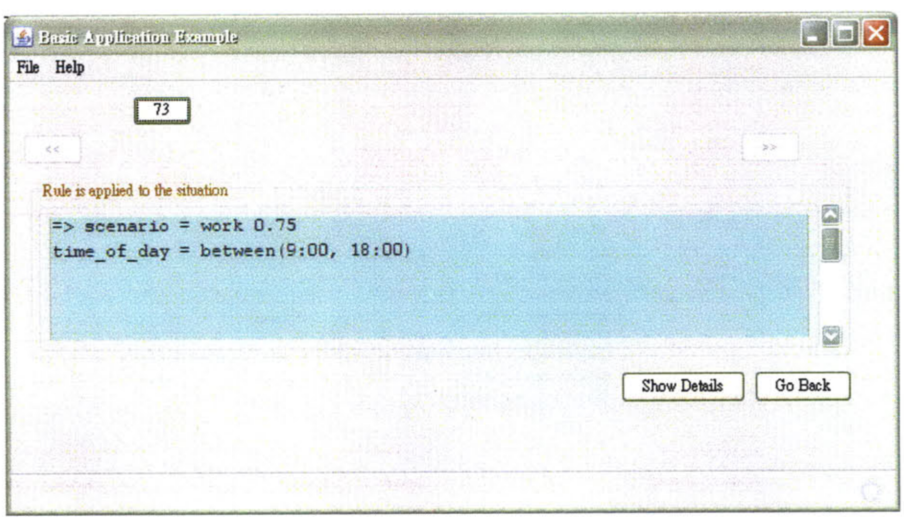

Figure 3-39: The rule selected (#73) was applied to the situation because the condition held.

11. The user decides to add a condition to rule #73 by clicking on the "Add Condition" button. The interface also allows the user to delete conditions or to edit existing conditions.

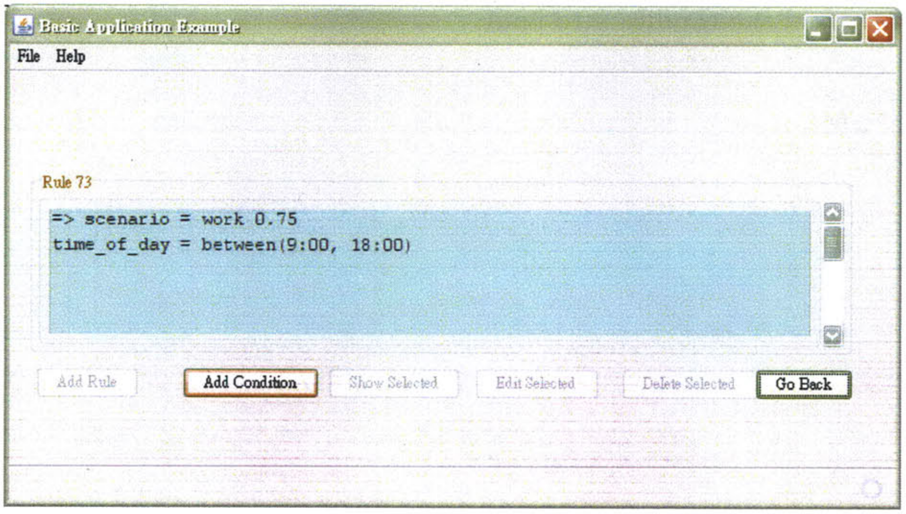

Figure 3-40: The user decides to add a condition to make the rule less general.

12. The user wants to make the rule less general by adding a condition related to the number of companions. The user then selects this attribute from the pull-down menu. The attributes are now sorted alphabetically; in the future, they can be grouped or sorted based on the category listed in Tables 2.1 to 2.10 to make the attribute selection more efficiently.

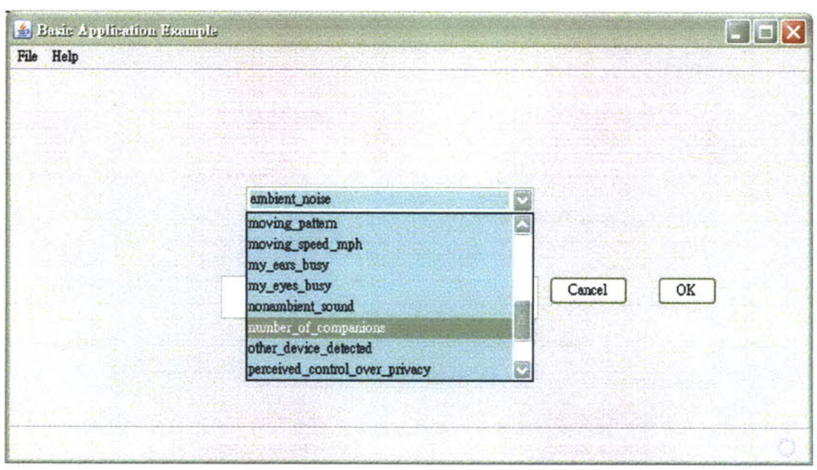

Figure 3-41: The user selects the attribute "`number_of_companions`" from the pull-down menu.

13. The user decides that the work scenario should hold when it is between 9am and 6pm **and** the number of companions is greater than 2. Hence a new condition: "`number_of_companions` >2" is added to the rule. The current interface requires user's prior knowledge of the legal values for each attribute.

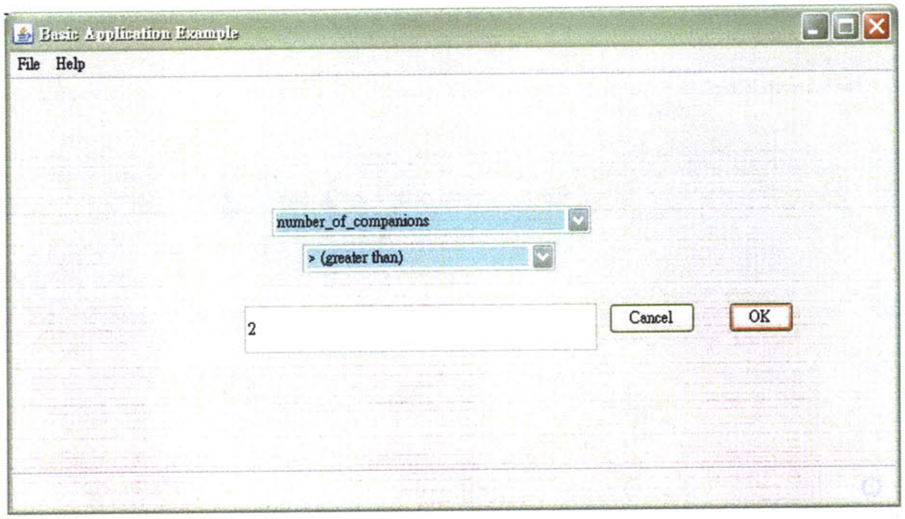

Figure 3-42: The user completes the new condition: "`number_of_companions` >2".

14. The rule has now been modified with the new condition "`number_of_companions` >2". (In addition to the pull-down menu, user is also allowed to modify any part of the rule in free text.)

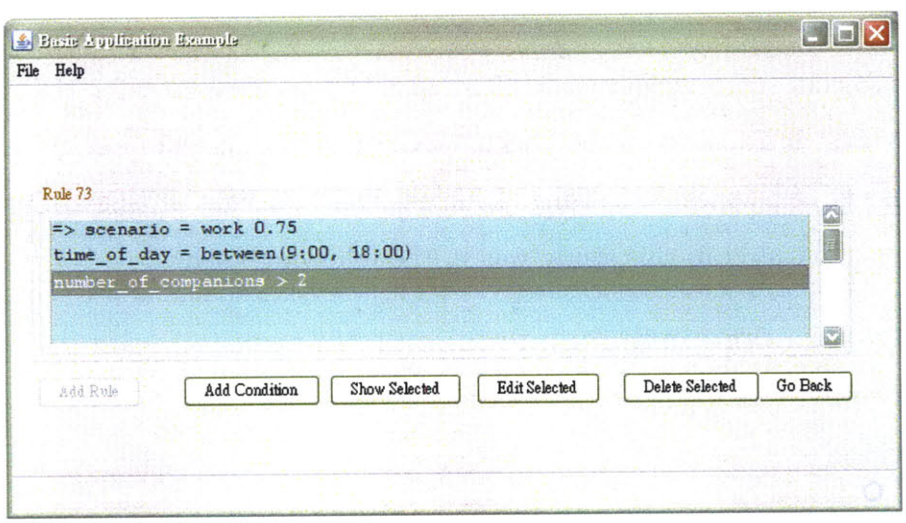

Figure 3-43: The rule has been modified with the new condition "`number_of_companions` >2".

15. The modified rule is saved in the system. The user has the option of running regression testing to see whether the modified rule set is able to correct the misbehavior.

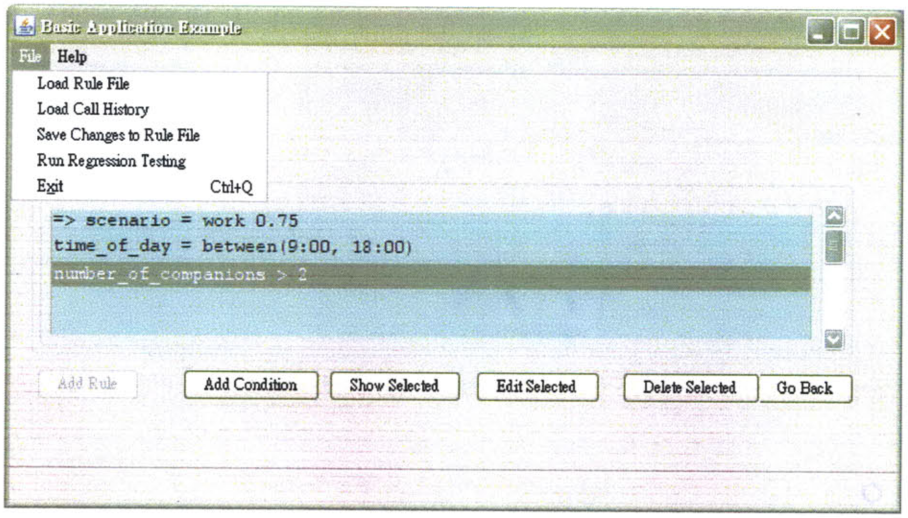

Figure 3-44: The modified rule is saved in the system.

3.2.2 User Study: Debugging Interface Study

Having built an interface for debugging, we would like to know whether general users are able to use such an interface to modify/create rules to correct the device when it misbehaves.

Five subjects (current/former MIT students) were recruited for the user study. They were first introduced to the vocabulary and syntax used in our system, as well as the concept of backward chaining. Different from the rule-writing study in section 3.1.5, the syntax used in this study was in the following form:

```
=> conclusion  certainty
   condition
   condition
   ...
   condition
```

Then, the subjects were shown two scenarios that most subjects in previous user studies considered as "misbehaving": (1) the device rings and flashes when the user is in class (2) the device rings when the user is in a meeting. For the rule set used in this study, one rule for each scenario was purposely "broken", so that the device running the rule set would misbehave. The subjects were asked to find out which rule was broken for each scenario and fix the rule, using the debugging interface.

None of the subjects had prior exposure to the vocabulary. Considering the cognitive load required to learn the vocabulary, the syntax, the concept of backward chaining, and the interface, the limited amount of time to complete two bug-fixing tasks, and the fact that there are several ways to "fix" a rule, we did not evaluate the subjects' production on the scene. As long as the subjects felt they had fixed the rules, the study session was completed.

Based on the results after the study, all 5 subjects were able to debug the knowledge base. While 3 subjects fixed the purposely broken rules by adding conditions, changing the certainty factors, changing the value of a particular attribute, or creating new rules that would suppress the "buggy" rule, the other 2 subjects modified other rules to their liking. Given that subjects did not have time to explore the entire rule set to find out what existing rules might be affected by their bug fixes, it would be too harsh to evaluate the successfulness of the bug fixes by running the decision-making system and see if the final recommendation for each scenario was corrected. By look at the subjects' production manually, the subjects who fixed the purposely broken rules were able to stop the broken rules from firing with the bug fixes; one subject who modified other rules was able to suppress the broken rule from firing, but the other subject could not.

All 5 subjects asked questions while navigating the rule set to debug. Most questions were interface-related, for example, "*What are these buttons?*", "*How do I view the details?*", etc. There were questions on a function that the current interface was incapable of performing (how to see all the rules related to a particular caller or calendar entry).

In addition to interface-related questions, there were questions and comments such as "*How can I set the phone to vibrate?*", "*I wanted to see whether I could shut the* *phone off ...*". There were also comments about not being able to keep track of the rules that had been previously visited. It seems that the subjects either did not have the concept of debugging, or were overwhelmed by the depth of the rule chain.

For users who understand the concept of debugging, there is certainly room for improvement for an interface that could better facilitate the debugging process. Before we start changing the interface, we first need to investigate what information should be presented to the user at the beginning of the debugging process: the facts that the device knew when the scenario took place? the one rule on the highest level that caused the misbehaved response of the device? the rules on the highest level related to the expected response? all the rules related to a particular piece of fact? or other pieces of information we have not found out from this user study?

For users who are overwhelmed by the depth of the rule chain, a big design question would be: how can we visualize large amount of data (*i.e.* the entire tree structure of the rule set) given a small screen real estate? The interface we created here apparently has not reached the goal. However, even with the "zoom in/out" function on some current models of smart phones, users still have to mentally keep track of items that can't be seen on the screen.

For users who do not have the concept of debugging, or who do not want to spend time debugging, there might be ways to obtain bug fixes through collaboration or sharing mechanisms. We will discuss the possibility in Chapters 4 and 6.

Chapter 4

Discussion

The previous two chapters described the steps of constructing a polite mobile device. We explored the signals available to a smart mobile device and identified the ones that are important to inferring when and how to interrupt the user when there is an incoming voice call or text message. We found the major factors that a device should be aware of in order to behave politely: the user's (aural and visual) attention, bystanders' (aural and visual) attention, and the user's physical availability. The rule set in our work was built to express these major factors by making inferences from the signals that can be measured or detected from the device. The vocabulary defined in the rule set is intended to capture the general users' definition of politeness with regard to handling voice calls or text messages. An exploratory study has shown that the vocabulary could be expanded to approximate the general users' definition of politeness, and we can look further into the impact on the behavior of the device when a new attribute is added to the vocabulary.

Our decision-making system determines how the device should respond to a call or message based on the information available to the device, and the rules used in the system can be understood by users with programming experience. In addition, users are able to use the vocabulary to come up with rules that meet their needs. When the mobile device misbehaves, users who are curious about the reasoning process of the system are able to fix the rule(s) by using the debugging interface we proposed, with the assistance in the interface elements and some prior exposure to the vocabulary and the concept of rule creation.

Based on our findings and observations, obviously there is still a gap between our current system and a product that can be distributed to the general public and used immediately. We would like to know how big the gap is by looking into the limitations in the domain itself, the technologies we have at the moment, and the questions regarding interface design for a mobile device.

4.1 Limitations

With the increasing computational power and number of sensors on a smart mobile device, it is natural for us to hope for a device that is, eventually, as polite as a human. However, the device is typically carried inside a pocket or a bag, or is sometimes placed in a location far away from its user. In other words, the device can more realistically be compared to a human being who is blindfolded and muffled, or a human assistant who is at a distance. The amount of information available to the device is thus limited.

Different from some relevant work done by others, we would like all the signals be gathered in a non-intrusive way: non-intrusive to the user, and non-intrusive to the world around the user. Anecdotally, several subjects in our user studies made the same comments about wanting the polite device to "monitor my dreams", but then immediately rejected the idea of being attached to sensors while sleeping. On the other hand, if the world around the user is carefully instrumented, it is very likely that the device will gather more information that will be helpful in determining when and how to interrupt its user. In reality, it is impossible to instrument every corner of the globe. That being said, given how pervasive and popular the WLAN-based and GPS technologies have become in the past few years, we are optimistic that several years from now, technology will enable us to collect more signals automatically (without purposely instrumenting the world), and the device will be able to make use of the signals for better inference.

4.2 Simulated *vs.* Real Signals

For our work, the signals are simulated, and the decision-making system is run on a computer, rather than on a smart phone. An attempt at collecting real signals with a mobile computing device to predict user interruptability was made in 2005 [11]. The attempt was unsuccessful due to several hardware issues: (1) the static generated by the device itself created noise in the audio data collected by the microphone, which adversely affected the quality of sound-related signals and the accuracy of speech recognition, (2) the GPS receiver could not be plugged into the device directly, (3) the battery power of the device restricted the duration it could be operated to collect signals, and so on.

With the advancement of smart phone design in recent years, one may argue that the hardware should no longer prevent us from collecting real signals with a mobile device. We were able to collect real signals for 2/3 of the attributes listed in Tables 3.2 and 3.3. However, we did not want to restrict our work to the technologies at present: to accurately predict the user's moving pattern from accelerometer readings, it requires a large amount of labeled training data [21]; to detect bluetooth-enabled devices in a room, precise calibration is required to measure the distance between devices; to detect the number of speakers in a room using speaker diarization, prior knowledge of the speakers (example speech from the speakers) or the linguistic information in the detected speech is required [39]. Vast research efforts have been devoted to these areas of studies; by using simulated signals in our work, we assumed that all the signals a device needs to know would be readily available in the near future, and we were able to focus on how a device would behave using our system, and how the system could be improved to make the task of constructing a polite device easier.

4.3 User Interface

In addition to how to present large amount of information on a small screen real estate, and how the visual interface element should be designed to facilitate the presentation and navigation of the rule set, another important design question is: is there an interface for users with different levels of interest in writing rules, programming, or debugging?

Based on the user studies we conducted, users who identified themselves as programmers generally had an easier time learning the vocabulary and writing rules (compared with non-programmers), and were more willing to spend time crafting the rules and debugging, in order to customize the device for their own needs. The non-programmers in our studies did show interest in creating rules, but most of them preferred not to invest too much time (over 10 minutes) in "teaching the phone everything"; in addition, some find the concept of "programming" or "debugging" intimidating. We believe that a carefully designed interface could encourage users to start creating rules or fixing rules without reminding them that this is a process equivalent to "programming" or "debugging". Alternatively, we can utilize social computing and collaboration mechanisms to distribute the effort of creating/modifying the rules and reporting the bugs.

Chapter 5

Related Work

Prior research efforts and existing applications have partially addressed the "when and how to interrupt the user" problem. Some have made use of sensors to infer the physical and social situation where a mobile device is embedded and whether the user is interruptable; some have helped the user prioritize incoming messages or schedule events based on calendar entries and user preferences. Others adaptively learn how each piece of information should be presented to the user based on user activities. In this chapter, the related work is reviewed from three aspects: (1) context-aware computing, (2) interruptability, and (3) privacy and social psychology.

5.1 Context-Aware Computing

Context-aware computing has been a concept subject to extensive research for decades. *Context* refers to the physical and social situation in which computational devices are embedded [2], which is essential information if the devices are to be smart and polite. Bardram and Hansen [2, 3] developed a WLAN-based prototype application, which was used in hospital for communication among medical personnel. In addition to WLAN-based location, physical objects used by medical personnel indicated their activities. Judging from the indicated activities, the application would facilitate social awareness and suggest courses of action that could be taken by the medical staff. For example, if a nurse has entered the "active zone" (where a patient is) and has picked up a medicine container, the nurse's current location and the medicine container indicate that the nurse might be busy administering the medicine, and thus might not respond to the doctor's paging immediately. In our work, we did not have a mechanism to detect what physical object is used by the user, but we did incorporate location information to infer the user's interruptability.

A context-aware application typically has to interact with its user in order to confirm that the context determined by the system is correct. With a proper internal model built with enough training data, such interaction can be done unobtrusively, without interrupting the user. For example, the personalized stock tracking application by Yoo et al. [42] unobtrusively gathered positive and negative feedback from its user by whether or not the user purchased the stock recommended by the application. When the recommendation was rejected by the user, the user could provide explicit feedback by clicking on a button next to the recommendation, and the feedback would change the internal model. In our work, the internal model was built with rules inspired by calendar entries and interviews with real users. If the decision-making system makes a wrong decision on how the device should respond, the user is free to use the debugging interface to modify the rule set, which will then change the internal model.

For other context-aware applications [28, 36, 7], various sensors, microphones, and cameras were used to create user context in the form of "who, what, where, when, how, and why"; machine learning techniques were used to predict user preference or to help user perform form-filling actions. In our work, we did not install any sensor or device in the surroundings of the user to create context. Our system collects signals and measurements from the sensors on the mobile device. The context is then inferred by a rule-based system using these signals, and the decision of how the device should respond is made based on the inferred context.

There are some commercially-available products that relate context-aware computing to the safe use of mobile devices: *DriveAssist* [1] and *iZUP* [18] are applications that prevent incoming phone calls and text messages from distracting the users when they are in motion, and the users cannot use the mobile phone while driving.

Key2SafeDriving [29, 30] is a bluetooth device embedded in a car key that turns off the user's mobile phone when the user is driving.

5.2 Interruptability

Attention is a precious resource. Whenever an agent (a person, a device, or an application) initiates an interaction with its user, the agent must first interrupt the user. However, interrupting a user's attention could significantly delay the task that the user is attending to [41], no matter whether the interruption is through personal visit, phone call, email, or other means of interaction. Moreover, as pointed out by [37] and [25], even when a piece of incoming information is potentially useful, the user may not always be available to pay attention to it; hence, a polite device is expected to evaluate both the interruptability of the user and the benefit of incoming information, in order to wisely initiate the interruption if the incoming information is worth the user's attention, or if the information could considerably benefit the user's current task.

In the case of a device handling incoming phone calls, it is very difficult to find out the urgency and the benefit of a call automatically until the communication between the caller and the user has started. In our work, the device allows the caller to indicate whether the call is urgent, work-related, etc. prior to requesting the user's attention; the indicated urgency will then be treated as a known fact and passed on to the decision-making system.

5.2.1 Coordinating Interruptions

When considering when a device should interrupt its user, we found that Mcfarlane [26] had pointed out four methods for an agent to coordinate interruptions:

1. The agent interrupts the user whenever a piece of information is received. This is the most commonly used approach by most phones and email notification software nowadays.

2. The user decides not to redirect their attention to the incoming information until the user becomes available.

3. The agent starts the interruption when the user seems available, or when the incoming information seems valuable.

4. The user specifies a fixed period of time that does not allow any interruption.

Based on the study conducted by Mcfarlane, there is no one "best" choice of method for coordinating interruptions, and the trade-offs between these methods lie in the nature of the incoming information and the current task that the user is attending to; the trade-offs also lie in the different expectations of user performance of the current task (*e.g.* precision, efficiency, etc.). We agree that there is no best choice among these four methods. In our work, we considered all four methods when creating the rules: depending on the urgency of incoming information, user-specified time frame, and the inferred user interruptability, the decision-making system would decide on the most suitable method to initiate the interruption.

The *Scope* system by [40] learns the models of prioritizing incoming information from user behaviors and from explicit user feedback. The system automatically assigns an urgency score to each piece of incoming information based on, for example, uppercased words in the subject line of email, sender, nature and number of recipients, content of the header and body, etc. The more urgent items would be placed more centrally on the visual interface. *Nomadic Radio* [32, 33] is an audio-only device worn around the user's neck. It determines user interruptability based on user actions and prioritizes text-based messages sent to the user, and the user is then notified with scalable audio cues. Our work focuses on devices that are typically placed in a pocket or backpack, and our system currently does not handle incoming email messages. For text messages, our system only checks the message sender to infer the urgency of information. In the future, we can incorporate the criteria used by *Scope* and *Nomadic Radio* to better determine the urgency of incoming information in the form of text.

Garblephone [34] is a device that allows the caller to gauge the activity level of the user by listening to the user's conversational state, while allowing the user to screen the calls. In our work, the device is expected not to initiate the interruption when the user seems to be uninterruptable. Our system reveals the user's location or availability to the caller when necessary, and the caller is given the opportunity to indicate the urgency of the call prior to initiating the interruption.

Marti and Schmandt [24] built a system that determines whether a phone call should be accepted by polling everyone who is in a conversation with the called party. The ones alerted by the system (with a finger ring that vibrates) vote on whether the conversation can be interrupted by the incoming call, without knowing whom the phone call is intended for. In our work, the user and those around the user are not meant to be interrupted in any fashion, if they don't seem to be interruptable; the interruptability is determined by signals available to the device and the inferred information from the signals, without the input from others around the user.

In [25], McCrickard and Chewar discussed using the attentive user interfaces (AUI) paradigm to model and adapt to a user's attentional state, and hence to bring the right information at the right time to the user. They proposed a framework that allows costs and benefits to be described, where "costs" refer to sacrificing the attention from other tasks the user is engaged in, and "benefits" refer to fulfilling the user's goals in aspects such as task comprehension, reaction to notifications, pacing interruptions, and satisfaction of the overall experience. It was suggested that designers of notification systems consider the AUI paradigm when trading off between diverting user attention and delivering timely notifications. Our current system does not compute the cost; it infers the interruptability of the user and others around from available signals. Our system does not compute the benefit of an interruption; it determines whether the user should be interrupted based on caller ID and caller-identified urgency.

5.2.2 Measuring User Interruptability

According to Hudson et al. [17], people tend to have a constant daily rhythm in the attitudes toward interruptions: for example, those who feel more productive at work in the morning may be less interruptable before lunch time. However, the attitudes may vary based on the current task a person is engaged in: if the current task is planned, the person is generally less interruptable than if the task is unplanned or spontaneous. If the task is a meeting, the interruptability is reversely correlated to the size of the meeting. That being said, a person's availability may change depending on the nature of the interruption, and the attitudes toward interruption typically involve the trade-off between wanting to avoid interruption and appreciating its usefulness. In our work the device doesn't know the usefulness of incoming information, but it is able to infer the user's attitude toward interruptions based on the user's calendar entry and the number of companions around the user.

The wizard-of-oz study by Fogarty et al. [9] used detailed events or situations in a single-person office with manually simulated sensors, which were categorized as occupant-related, guest-related, and environment-related. The result shows that people often have strong feelings about particular times of day being "obviously not interruptable", but often have more ambivalent attitudes towards "partially interruptable" times. Their study used the machine learning approach to assess whether "now is a bad time" to interrupt the user. In our work, the results from user studies also show that there are particular times of day that is not interruptable (*e.g.* late at night, in the morning before starting to work), and the times are specified in our rule set using the "time of day" attribute.

Horvitz et al. [14, 15] have developed systems that sense signals about user attention, and these signals are shared with other users to imply interruptability. Our system utilizes some of the signals used by Horvitz et al., and they are incorporated into the vocabulary of the rules for the user to describe her/his state of interruptability. These signals include: ambient noise, user presence, readings from accelerometers, calendar information, motion of devices, and location sensing via GPS signals.

Horvitz et al. have also built models to predict the cost of interrupting the user. In their studies, the cost of interruption was based on self-reported mapped dollar amount that the user was willing to pay to avoid each interruption. Measuring user interruptability in this way seems better than the approach used in [17], where a device would interrupt the subjects of the study and present them with a 8-question survey on their current interruptability. The system in our work does not measure or predict the cost of interruption. Instead, the inferred information about the aural/visual attention of the user (and of others around the user) is used as an indication of whether and how the interruption should be initiated.

Ho and Intille [13] proposed that the transition between user motions could be an indication of transition between tasks, and that no active motion for a long period of time could be an indication of "having nothing to do". Hence, the user would seem to be more interruptable either when a transition between motions is detected, or when the user appears to be "motionless" for a long while. However, it is a complicated problem to determine whether the lack of motion indicates "having nothing to do", as there are situations where a doctor being seated is talking with patients, an attorney being seated is talking with clients, or a chemist standing still is conducting an experiment with a flask or beaker in hand. Despite the fact that a smart mobile device is equipped with acceleration and orientation sensors and is capable of detecting various moving patterns, we acknowledge that this is a hard problem, because more context information is required to determine whether the user "has nothing to do" and can be interrupted.

Bernstein et al. [5] used multiple sensors (accelerometer, potentiometer, and microphone) to predict user interruptability. Although the data for their study stemmed from only one subject, they were able to point out that scenarios the user is in are important to interruptability prediction. In a different fashion, Chen et al. [6] created a task-independent model of interruptability based on user's physiological state (*e.g.* heart rate variability and electromyogram) and found out that the mental load of the user contributes more to such a model than the user's muscle activity. In our work, we do not attach any sensors to the user to measure physiological state. We acknowledge that having a "`scenario`" attribute in the rule set makes it easier for general users to understand how the rules are constructed. However, different scenarios often share similar properties such as the concerns about user's aural/visual attention, bystanders' aural/visual attention, and user's physical availability, etc. The rules in our system were created to express these common properties, rater than describing each scenario independently.

Kern and Schiele [21, 20, 22] proposed the notion of *social interruptability* and *personal interruptability.* Multiple sensor streams were used (3D body-worn acceleration sensors, microphone, and WLAN-based location sensor) to create 50 "low-level contexts" expressed by acceleration, location, and audio information. Unlike other studies mentioned previously, the subject wearing the sensors did not have to report his interruptability. Instead, situational videos of the subject were made and annotated with sensor measurements; other subjects in the study were asked to watch each video and report their interruptability as if they were under such situation. In this way, the subject being measured would not be interrupted, and the system would learn the prediction model based on sensor measurement and the objective feedback from subjects not wearing the sensors. In our user studies, the subjects were only given text descriptions of the scenarios. Although we did not ask the subjects to wear any sensors, the low-level rules in our system do make use of signals such as acceleration, location, and audio information.

5.2.3 Existing Applications

Various applications have been developed in the past decade to address the preference for interrupting the user to the minimal extent. For example:

OwnTime [31] is a timespace management system that attempts to allow flexible meeting scheduling with minimal user interruption. The interface provides a subtle notification when a meeting request is received, and the interface fades away if the user does not respond for a certain period of time. The system has shown the potential for minimally disrupting the user's current task. In our work, we have "beep2" as one of the output modalities for the device, which requests the user's attention in a gentle way. Our system also casts part of the responsibility of being polite to the caller when it is unsure about the benefit of interrupting the user.

Locale [4] is an application for a mobile device. It automatically determines the user's location (by GPS, Wi-Fi, and GSM positioning) and changes the ringer volume setting based on user-preset values. In our work, location is an important piece of information that indicates the current activity of the user. In addition to GPS and Wi-Fi signals, our system also accesses the user's calendar entry to infer where the user might be.

Cell phone manufacturer HTC has announced a feature named *HTC Sense* [16]. A device running *HTC Sense* is able to automatically adjust the volume of the ringer depending on whether the device is in a bag/pocket or has been picked up; if the device is then flipped over, it is a gesture of silencing the device. In our work, the decision-making system determines whether or not the user should be interrupted prior to initiating any form of interruption. Hence, a device in a bag will not ring if the system has decided not to interrupt the user by making a noise, and this saves the user from having to reach for the device, take it out of the bag, and flip it over.

5.3 Privacy and Social Psychology

The level of user-perceived privacy can be determined by multiple factors: physical space, property of incoming information, size of display device, and level of perceived control (over a piece of information or a situation). The research efforts mentioned in this section have provided the basis and inspiration for our work.

Kaya and Weber [19] looked into the relationship between the level of privacy and the physical space. They defined *density* as the "physical condition involving space limitations" and *crowding* as "subjective, psychological experience that is associated with a feeling of lack of control over the physical environment". If the level of crowding decreases then the achieved level of privacy increases, while the density of the physical space may remain the same. Our current system has yet to be capable of detecting or inferring the level of crowding. Further research can be conducted to determine how crowding can be expressed from a mobile device's point of view.

As for the type of incoming information, Häkkilä and Chatfield [10] discovered that SMS (text) messages are perceived as more private than normal calls. They also discovered that text messaging is not comparable with any other form of electronic communication (for example, emails, which are commonly forwarded without the sender's consent), but are more comparable with traditional letters. Although our work mainly focuses on handling voice calls, a device should handle text messages with more concern about privacy than other forms of incoming information.

The "western models of privacy" mentioned in Little et al. [23] serve as a good reference to distinguish different forms of user-perceived privacy:

- Physical: how physically accessible a person is to others
- Psychological: a person's right to decide whom to share personal information with
- Social: control social interactions by controlling distance between people
- Informational: a person's right to reveal personal information to others

Spiekermann [38] investigated the (perceived) privacy and the (perceived) control in ubiquitous computing environments and found that "it is really more about perceived control than it actually is about the end state of privacy itself". In Spiekermann's research, "privacy" is defined as the control a person has over information about himself or herself; if one does not feel competent enough to master a situation, s/he will not feel in control. Similar to devices in ubiquitous computing, a polite device should carefully consider how much information to reveal to the caller/sender, and how to protect user privacy when presenting the incoming information to the user.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We investigated the human definition of "politeness" in the context of handling voice calls and text messages in mobile devices and obtained a first approximation. We also found the common properties shared by the scenarios where most users would find embarrassing or irritating when the device initiates the interruption with the wrong timing or wrong modality: (1) the user's aural/visual attention is not to be disrupted (2) the aural/visual attention of the bystanders (others around the user) is not to be disrupted (3) the user is not physically available to respond to the call. We interviewed subjects and examined the vocabulary used in our system to describe the scenarios and how a device could learn how to behave. At the same time, we explored the signals that would be good for a device to know and identified those important to determining when and how to interrupt the user. We compared the vocabulary used by the subjects and the one in our system, and the 42% overlap suggests that we have captured some of the user intention, but there is still room for improvement.

We built a rule-based decision-making system, which infers user interruptability from the above-mentioned common properties using the information available to the device, and then determines how the device should respond to an incoming call or message. The current size of the rule set is about 250 rules. And 2/3 of the rules in our system could be understood perfectly by at least half of the programmers in our user study without any assistance.

With a minimal amount (30 minutes) of introduction to the vocabulary, syntax, and the basics of rule writing, almost all subjects in our user studies were able to create rules for their own needs, using the vocabulary defined in our work. Among the 37 rule-writing attempts by programmers, 34 of them (92%) had over 50% of vocabulary overlap with the rule set in our system; particularly, 19 of them (51%) had 100% vocabulary overlap. As for the 25 attempts by non-programmers, 17 of them (68%) had over 50% of vocabulary overlap. The existing rule set in our work has a decent vocabulary coverage of scenarios where a device is expected to behave.

To make the system more accommodating to individual users' needs, we created a debugging interface allowing users to modify existing rules and create new rules. The major non-interface issue observed from our user study is the unawareness of the concept of "debugging". Given the fact that most users are accustomed to the existing mobile devices on the market, their initial reaction to a misbehaving device tends to be how to turn off the device, or how to force the change of the device's output modality, rather than figuring out why the device (mis)behaved as such, or how come the device did not follow what it had been told to do. That being said, when subjects in our user study were asked to debug the rules to correct the behavior of the device, all of them were able to modify the rules using our debugging interface, with the introduction to the vocabulary and syntax, and with the assistance on interface navigation.

Another major issue observed from our user study is the difficulty of keeping track of the structure of the rule set. We see this as an interface design problem beyond selecting and arranging interface elements – it is a problem about how to present large amount of information on a small screen real estate. With the current capability of a smart mobile device, it is not a difficult task to implement the "zoom in/out" function for the users to control the amount of information being displayed on the screen. However, when the users have to navigate a complex tree structure to debug the rule(s) in a rule set, it seems unavoidable that they have to mentally keep track of nodes and branches that have been previously visited but are not displayed on the screen at the moment. At this point we do not have a solution to this problem, but we do see this as an opportunity for further research in the area of interface design for small form factors.

6.2 Future Work

There is still room for improvement for our work to become a product that can be released to the general public. Future work can be addressed from the following three aspects.

6.2.1 When more signals on the wishlist are granted

With the advancement of technology, we do expect more signals will be available to a smart mobile device. As more signals become available, experiments can be done to see whether augmenting the vocabulary with new signals and rewriting the rules will help the device make better decisions.

6.2.2 User demographics

1. In our user studies, most subjects who are Blackberry users expected the device in our work to behave differently when the device is plugged into the charger. This is a gesture we were unaware of when we defined the vocabulary for our rule-based decision-making system. Some Blackberry users also expressed the need of prioritizing email messages from different accounts, on top of text messages and voice calls. Before one starts modifying the vocabulary and the rules for these anecdotal requests, it will be helpful to investigate the use habits of users of different smart phone models and evaluate the importance of incorporating certain features and use habits into the rules.

2. In some cultures, the output modality of a mobile device is restricted by the airtime or power available. In some other cultures, the use of a mobile device is strictly prohibited in a certain locations. Currently, our work allows users

to create new rules if the existing rules are insufficient. In the future, rules for different cultures can be created in advance and be applied according as soon as the device finds out its geographical location, so that the device will automatically conform to the local etiquette and prevent its user from committing a faux pas.

6.2.3 A better debugging interface

1. According to our user study on the debugging interface, different users seemed to have different requests on what information to be presented at the beginning of the debugging process. To make the debugging process easier, one has to find out how to present the relevant information to the users without making them feel overwhelmed or confused.

2. For users who have no prior experience or interest in debugging, one can investigate how to design an interface that allows the users to customize the rules without making them feel the pressure of having to program.

3. Along the same line, it will be interesting to look into how much does a user need to know about the rules, the decision-making system, and the debugging. If a user has no interest in finding out the root cause of why the device misbehaved and only wants a "quick fix", is there any way to make the phone behave politely without the user fixing the bug? One possible approach is to make use of social computing and remote collaboration to distribute the effort of bug reporting and bug fixing among multiple users. To figure out the most efficient approach, experiments can be conducted on connecting users, categorizing bugs, creating ownership and resolving editing conflicts, mechanisms for selectively updating the rule set, building a sound reward system, etc.

4. Last but not least, an important question that remains to be answered is: how shall we represent large amount of data on a small screen real estate? How shall we design an interface that allows the user to navigate through complex rule

structure, without the burden of memorizing information that is not visible on the screen? We are curious to find out the challenges and what it will take to possibly solve the problem.

Bibliography

[1] AegisMobility. Driveassist. `http://www.aegismobility.com`, 2008.

[2] Jakob E. Bardram. Applications of context-aware computing in hospital work: examples and design principles. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1574–1579, New York, NY, USA, 2004. ACM.

[3] Jakob E. Bardram and Thomas R. Hansen. The aware architecture: supporting context-mediated social awareness in mobile cooperation. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 192–201, New York, NY, USA, 2004. ACM.

[4] Clare Bayley, Carter Jernigan, Jasper Lin, Jennifer Shu, and Christina Wright. Locale. `http://androidlocale.com`, 2008.

[5] Abraham Bernstein, Peter Vorburger, and Patrice Egger. A scenario-based approach for direct interruptability prediction on wearable devices. *International Journal of Pervasive Computing and Communications*, 3(4):426 – 438, 2007.

[6] Daniel Chen, Jamie Hart, and Roel Vertegaal. Towards a physiological model of user interruptability. In *INTERACT (2)*, pages 439–451, 2007.

[7] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity sensing in

the wild: a field trial of ubifit garden. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1797–1806, New York, NY, USA, 2008. ACM.

[8] Randall Davis, Bruce G. Buchanan, and Edward H. Shortliffe. Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, 8(1):15–45, 1977.

[9] James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction*, 12(1):119–146, 2005.

[10] Jonna Häkkilä and Craig Chatfield. 'It's like if you opened someone else's letter': user perceived privacy and social practices with SMS communication. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 219–222, New York, NY, USA, 2005. ACM.

[11] Johan Hesselberg. Estimating user interruptability using contextual parameters and machine-learning algorithms. Master's thesis, Royal Institute of Technology, Stockholm, Sweden, 2005.

[12] Ken Hinckley and Eric Horvitz. Toward more sensitive mobile phones. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 191–192, New York, NY, USA, 2001. ACM.

[13] Joyce Ho and Stephen S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 909–918, New York, NY, USA, 2005. ACM.

[14] Eric Horvitz and Johnson Apacible. Learning and reasoning about interruption. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 20–27, New York, NY, USA, 2003. ACM.

[15] Eric Horvitz, Carl Kadie, Tim Paek, and David Hovel. Models of attention in computing and communication: from principles to applications. *Communications of the ACM*, 46(3):52–59, March 2003.

[16] HTC. HTC Sense. `http://www.htc.com/www/htcsense/index.html`, 2009.

[17] James M. Hudson, Jim Christensen, Wendy A. Kellogg, and Thomas Erickson. "I'd be overwhelmed, but it's just one more thing to do": availability and interruption in research management. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 97–104, New York, NY, USA, 2002. ACM.

[18] iZUP. iZUP. `"http://www.getizup.com/what_is_izup/faq"`, 2010.

[19] Naz Kaya and Margaret J. Weber. Cross-cultural differences in the perception of crowding and privacy regulation: American and Turkish students. *Journal of Environmental Psychology*, 32(3):301–309, 2003.

[20] Nicky Kern, Stavros Antifakos, Bernt Schiele, and Adrian Schwaninger. A model for human interruptability: Experimental evaluation and automatic estimation from wearable sensors. In *ISWC '04: Proceedings of the Eighth International Symposium on Wearable Computers*, pages 158–165, Washington, DC, USA, 2004. IEEE Computer Society.

[21] Nicky Kern and Bernt Schiele. Context-aware notification for wearable computing. In *7th IEEE International Symposium on Wearable Computing*, pages 223–230. IEEE Press, Oct. 2003.

[22] Nicky Kern and Bernt Schiele. Towards personalized mobile interruptibility estimation. In Mike Hazas, John Krumm, and Thomas Strang, editors, *Location- and*

Context-Awareness, Second International Workshop (LoCA'06), volume 3987 of *LNCS*, pages 134–150, Berlin Heidelberg, Germany, May 2006. Springer.

[23] Linda Little, Pam Briggs, and Lynne Coventry. Public space systems: designing for privacy? *International Journal of Human-Computer Studies*, 63(1-2):254–268, 2005.

[24] Stefan Marti and Chris Schmandt. Giving the caller the finger: collaborative responsibility for cellphone interruptions. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1633–1636, New York, NY, USA, 2005. ACM.

[25] D. Scott McCrickard and C. M. Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM*, 46(3):67–72, 2003.

[26] Daniel Craig Mcfarlane. Coordinating the interruption of people in human-computer interaction. In Angela Sasse and Chris Johnson, editors, *INTERACT'99: 7th IFIP Conference on Human-Computer Interaction Incorporating HCI*, pages 295–303, Amsterdam, 1999. IOS Press, IFIP TC. 13.

[27] George A. Miller, Christiane Fellbaum, Randee Tengi, Helen Langone, Adam Ernst, and Lavanya Jose. Wordnet. `http://wordnet.princeton.edu`.

[28] Yoosoo Oh and Woontack Woo. A unified application service model for ubihome by exploiting intelligent context-awareness. In *UCS*, pages 192–202, 2004.

[29] Jay Przybyla and Xuesong Zhou. Cell phone use while driving: a literature review and recommendations. "`http://www.civil.utah.edu/~zhou/cell_phone_and_distracted_driver.pdf`", December 2008.

[30] Jay Przybyla and Xuesong Zhou. Key2safedriving. `http://www.key2safedriving.net`, December 2008.

[31] Roy Rodenstein, Gregory Abowd, and Richard Catrambone. Owntime: a system for timespace management. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 200–201, New York, NY, USA, 1999. ACM.

[32] Nitin Sawhney and Chris Schmandt. Nomadic radio: scaleable and contextual notification for wearable audio messaging. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 96–103, New York, NY, USA, 1999. ACM.

[33] Nitin Sawhney and Chris Schmandt. Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments. In *ACM Transactions on Computer-Human Interaction*, volume 7, pages 353–383, New York, NY, USA, 2000. ACM.

[34] Chris Schmandt, Jang Kim, Kwan Lee, Gerardo Vallejo, and Mark Ackerman. Mediated voice communication via mobile ip. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 141–150, New York, NY, USA, 2002. ACM.

[35] Kjeld Schmidt. The problem with "Awareness": Introductory remarks on "Awareness in CSCW". *Comput. Supported Coop. Work*, 11(3):285–298, 2002.

[36] Anil Shankar, Sushil J. Louis, Sergiu Dascalu, Linda J. Hayes, and Ramona Houmanfar. User-context for adaptive user interfaces. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pages 321–324, New York, NY, USA, 2007. ACM.

[37] Timothy Sohn, Kevin A. Li, William G. Griswold, and James D. Hollan. A diary study of mobile information needs. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 433–442, New York, NY, USA, 2008. ACM.

[38] Sarah Spiekermann. Perceived control: Scales for privacy in ubiquitous computing environments. In *10th International Conference on User Modeling*, Edinburgh, Scotland, July 2005.

[39] Sue Tranter and Douglas Reynolds. An overview of automatic speaker diarisation systems. *IEEE Transactions on Audio, Speech and Language Processing V Special Issue on Progress in Rich Transcription*, 14:1557–1565, October 2006.

[40] M. van Dantzich, D. Robbins, E. Horvitz, and M. Czerwinski. Scope: providing awareness of multiple notifications at a glance. In *In Proceedings of the 6th International Working Conference on Advanced Visual Interfaces (AVI) 02*. ACM Press, 2002.

[41] Rini van Solingen, Egon Berghout, and Frank van Latum. Interrupts: Just a minute never is. *IEEE Software*, 15(5):97–103, 1998.

[42] Jungsoon Yoo, Melinda Gervasio, and Pat Langley. An adaptive stock tracker for personalized trading advice. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 197–203, New York, NY, USA, 2003. ACM.