Evaluating Intelligibility Usage and Usefulness in a Context-Aware Application

Brian Y. Lim and Anind K. Dey

Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213
{byl, anind}@cs.cmu.edu

Abstract. Intelligibility has been proposed to help end-users understand context-aware applications with their complex inference and implicit sensing. Usable explanations can be generated and designed to improve user understanding. However, will users want to use these intelligibility features? How much intelligibility will they use, and will this be sufficient to improve their understanding? We present a quasi-field experiment of how participants used the intelligibility features of a context-aware application. We investigated how many explanations they viewed, how that affected their understanding of the application's behavior, and suggestions they had for improving its behavior. We discuss what constitutes successful intelligibility usage, and provide recommendations for designing intelligibility to promote its effective use.

Keywords: Context-Awareness, Intelligibility, Explanations, User Study.

1 Introduction

Context-aware applications use implicit sensing and complex inference to automatically and calmly adapt for users [2]. End-users may not be aware of what these applications know, and struggle to understand and trust their behaviors [11]. To counter this, context-aware applications should be intelligible by providing explanations of their behavior [4]. Indeed, there have already been several context-aware applications that support some level of intelligibility (e.g., [1, 14, 15]). These systems support a limited set of explanations users can ask for: What, Certainty, Inputs, Why, and Why Not. However, Lim & Dey [5] found that users ask a wider range of questions of context-aware applications, and that different explanations have different impacts on user understanding. To support this wider range of explanations, Lim & Dey [7] designed Laκsa, which provides explanations to 8 question types for several context types. While that work provides a crucial step for designing intelligibility to be more usable and interpretable, it stopped short of evaluating the impact of intelligibility on users. Lim & Dey [8] investigated the impact of intelligibility on understanding and impression, but this was studied with questionnaires and 'paper' prototypes rather than an interactive prototype. Furthermore, intelligibility was shown "always on" to participants, so they were biased to look at the explanations. This leaves open the research questions: even if intelligibility can improve user understanding and trust, will users

adfa, p. 1, 2013.

want to use it, and, if so, how much? Moreover, given how much they do use, how much will that improve their understanding of context-aware applications?

Related work has explored the impact of explanations on end-users as they used context-aware systems. Tullio et al. [14] evaluated an intelligible interruption door display over six weeks, and found that users were able to "attribute concepts of machine learning to their system," but had difficulty remembering relevant features. Cheverst et al. [1] deployed the Intelligent Office System that provided explanation visualizations of rules and confidence. However, regarding explanations, their evaluation focused on eliciting user preference about visualization format, and not on their impact. Vermeulen et al. [15] conducted a pilot user study of PervasiveCrystal in a simulated museum with five participants, who "were able to use the questions interface to find the cause of events" of three tasks. We add to this body of work evaluating intelligible context-aware systems by explicitly measuring intelligibility usage in a high-fidelity prototype that provides over 9 explanation types (e.g., Certainty, Why, Why Not, What If) for three context types (Availability, Place, Sound). We also investigate the impact of this usage on user understanding of the application's inference. Our contributions show intelligibility is useful by investigating:

1. How much participants *use* intelligibility in a real context-aware application,
2. Their *opinion* of the *usefulness* of the explanations to understand application behavior and situations, and
3. How *useful* their use of intelligibility is on understanding and handling of these situations.

The rest of the paper is organized as follows: we describe an intelligible context-aware prototype which we developed for this study. Next, we elaborate on the quasi-field experiment we conducted, where participants engaged in "everyday" scenarios *in-situ* using the prototype. Following this, we present results of how participants used intelligibility and how that improved their understanding of application inference. Finally, we discuss design implications due usage patterns and constraints, and how to encourage users to use more intelligibility to further improve their understanding.

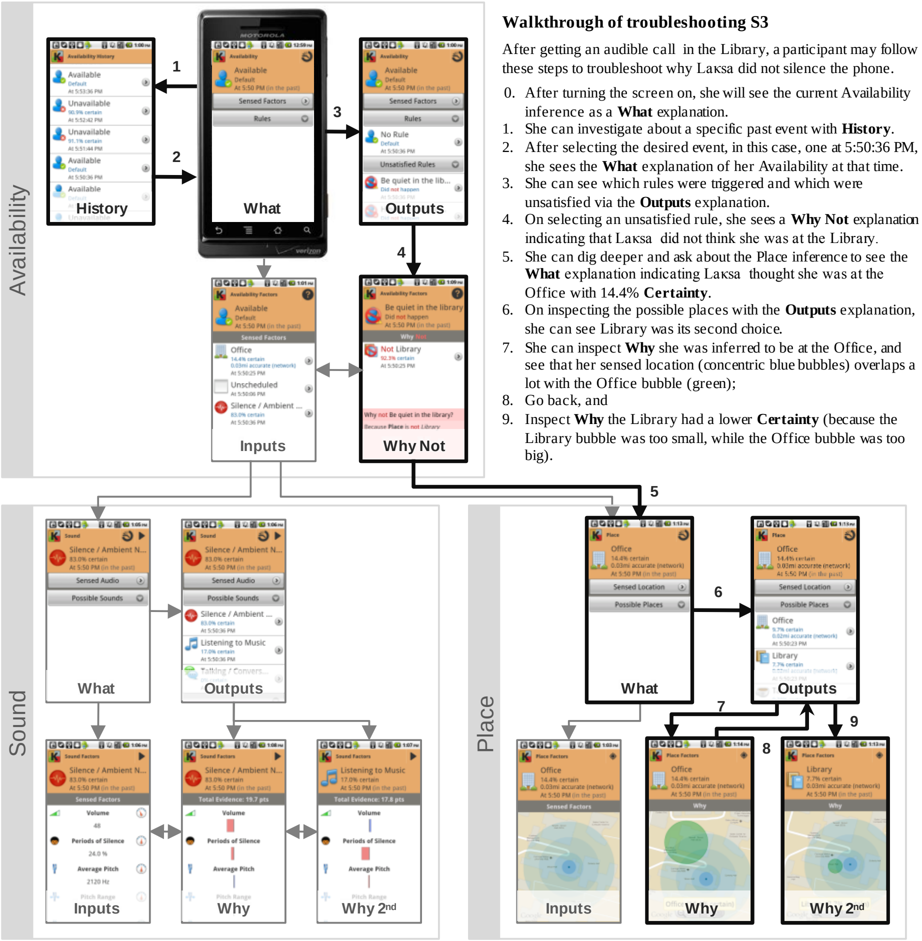
2 Laκsa 2 Prototype

Mobile phones allow people to keep in touch with others and be easily reachable. However, there are inappropriate times to receive calls, as they can be socially disruptive (e.g., in meetings and movie theatres), or they interrupt productive work. We have developed Laκsa, a mobile application which can automatically change the phone's ringer mode (e.g., [12]). It senses and infers the following contexts:

- **Availability:** *Available, Semi-Available, Unavailable* — is inferred from rules regarding the following three factors.
- **Place:** *Office, Café, Library, etc.* — is inferred by sensing latitude and longitude and matching to a pre-specified named place. The user's sensed location is modeled as a radial Gaussian, with decreasing likelihood further away from the latitude

and longitude coordinates (as in [7]). Each Place is inferred with different certainty based on how much the user's estimated location area "overlaps" with the circular area of the named place: more overlap leads to higher certainty.

- **Sound:** *Talking, Music, and Ambient Noise* — is inferred using a naïve Bayes classifier on features extracted from the phone microphone. Features extracted are similar to [9]: e.g., mean of power, low-energy frame rate, spectral flux, and bandwidth. These are renamed to lay terms that end-users can understand. The machine learning was implemented with an Android port of Weka [16].
- **Schedule:** *Personal, Work, Unscheduled* or *Other Event*.



Walkthrough of troubleshooting S3

After getting an audible call in the Library, a participant may follow these steps to troubleshoot why Laksa did not silence the phone.

0. After turning the screen on, she will see the current Availability inference as a **What** explanation.
1. She can investigate about a specific past event with **History**.
2. After selecting the desired event, in this case, one at 5:50:36 PM, she sees the **What** explanation of her Availability at that time.
3. She can see which rules were triggered and which were unsatisfied via the **Outputs** explanation.
4. On selecting an unsatisfied rule, she sees a **Why Not** explanation indicating that Laksa did not think she was at the Library.
5. She can dig deeper and ask about the Place inference to see the **What** explanation indicating Laksa thought she was at the Office with 14.4% **Certainty**.
6. On inspecting the possible places with the **Outputs** explanation, she can see Library was its second choice.
7. She can inspect **Why** she was inferred to be at the Office, and see that her sensed location (concentric blue bubbles) overlaps a lot with the Office bubble (green);
8. Go back, and
9. Inspect **Why** the Library had a lower **Certainty** (because the Library bubble was too small, while the Office bubble was too big).

Fig. 1. Screenshots of Laksa showing several explanation types of the upper-tier context Availability, and lower-tier contexts Sound and Place. Arrows between each screenshot shows how a user can transition from one explanation to another. The bold trace indicates how one may view explanations in 9 steps to troubleshoot Scenario 3 after the phone rang in the Library. The Intelligibility UI was adapted from [7, 8] using the "bubbles" metaphor to explain how Place is inferred, and the "weights of evidence" bar charts to explain feature votes for different Sounds.

Laκsa is *intelligible* [5] to help users to understand what it knows and how it makes inferences. Using the Intelligibility Toolkit [6], it provides explanations to questions:

1. **What** is the inference for the context? With how much **Certainty**? **When** was this value inferred?
2. **History**: what was the inference at time *H*?
3. **Inputs**: what details affect this context? (Factors, related details, etc.)
4. **Outputs**: what values can this context be inferred as? With how much **Certainties** are these values inferred?
5. **Why** was this value inferred?
6. **Why Not (Alt)**: why wasn't this inferred as *Y*, instead?
7. **What if** the factors are different, what would this inference be? (Requires user manipulation; only provided for Availability)
8. **Description**: meaning of the context terms and values.
9. **Situation** of what was happening to affect the inference to provide a ground truth of what was being inferred, e.g., playing an audio clip of what was heard.

We developed Laκsa for Android 2.2, and deployed it on the Motorola Droid for the user study. Sensing and inferencing were performed using background services on the phone every 30 seconds. Fig. 1 shows several screenshots of the Laκsa prototype. Users can transition from one explanation *page view* to another by clicking on buttons, option menu items, and flinging (swiping).

3 Scenario-Driven Quasi-Field Study

We explored intelligibility usage with a controlled scenario-driven user study to (i) present participants with critical incidences, and (ii) observe and measure their subsequent behaviors. We employed a quasi-field design (similar to 13] where each participant was brought to the necessary places to engage in various activities. The experimenter enacted critical incidences (e.g., by calling the participant's phone), and presented a printed flash card describing what was happening. The participant could interact with Laκsa as much or as little as she wished. After each scenario, the experimenter interviewed the participant asking about her opinion of the situation and the application, her understanding of how Laκsa made inferences, and how she may improve its behavior.

3.1 Scenarios

We employed four scenarios to span three situational dimensions: (i) Exploration / Verification (S1) of Laκsa's functionality and explanations; (ii) Fault Finding (S2, S3) to diagnose Laκsa's inappropriate behavior; and (iii) Preemptive Exploration (S4) where participants investigated a potential future situation.

S1: Talking in the office. The participant learned about and freely explores Laκsa's core features and explanations as Laκsa infers the Office location and Talking sounds.

S2: Missed call while reading and listening to music. The participant read a news article of her choosing from www.cnn.com while she listened to a song (a mostly vocal version of Sound of Silence) through speakers. Meanwhile, she missed multiple calls from a coworker because Laksa inferred the Music as Talking and automatically silenced its ringer. At the end of the song, the phone finally rang audibly. The participant learned that her coworker was frustrated from trying to call her repeatedly.

S3: Phone interruption in the library. The participant walked to a nearby library to search for a specific book to read. Meanwhile, a coworker called her phone, but Laksa misinferred the participant's Place as still in the Office instead of Library, allowing the phone to ring audibly in the quiet library.

S4: Preemptively checking availability in café. The participant received a flash card describing that she frequents a nearby café, and should check whether she will be able to receive calls there. She was not prompted what to do to achieve this objective.

3.2 Measures and Data Preparation

We measured how useful intelligibility was for the participants in terms of how much they used, and how that impacted their understanding of Laksa and its issues.

Usage of Intelligibility. We logged when participants viewed each explanation page in the UI. For each scenario, we measured which explanation types each participant viewed, how many (**# Explanation Types**), when they were viewed, for how long (**Duration**), and how often (**View Count**). We built a network graph for each participant scenario to illustrate the *sequence diagram* of how he used intelligibility (e.g., Fig. 1). As a measure of a usage pattern of intelligibility, we compute the **Context Ratio** of how many explanation types of deeper contexts (Place and Sound) were viewed compared to that of the shallower context (Availability). Next, we use these metrics to investigate their influence on user understanding.

User Understanding and Suggestions for Control. We coded transcripts into units of *beliefs* to characterize participant mental models about their understanding, using a coding scheme counting whether the participant indicated knowledge of the inferred **value** (e.g., Sound=Music), **alternative values** (e.g., P01S2 *"Talking (evidence=85.4) very close to music (84.?). Could have gone any way."*), inference **certainty** (e.g., P02S3: *"It was 9.3% certain I was at the Office"*), **inputs** (e.g., Pitch, Periods of Silence; *"the blue bubble was directly over the Library building."*), inference **model** (e.g., P17S3: *"...since the library bubble was very small then it calculated the probability was very low."*), **technical** details (e.g., P18S3: *"It seems to be based on its Wi-Fi connection, and ... because it said networking and it gave the location badly and we're deep inside a bunch of concrete and metal, so the GPS shouldn't be working right now."*), and **situation justification** (e.g., P02S2: *"The music was much more mellow, and they were really singing"*). We calculate an **Understanding Score** for each participant scenario by adding all 7 codes for both Place and Sound (Max=14).

Another measure of how well participants understood Laksa is how many effective *control suggestions* they provided to overcome any issues or problems in the scenarios. We calculated a weighted **Control Score** with a coding scheme counting whether the participant suggested **availability rules** (e.g., delete rule "Someone's Talking"), changing settings for inferring **Place** or **Sound**, and whether to change their own **behavior** (e.g., lowering the music volume). This score represents the number and effectiveness of suggestions provided for the scenario. Partially effective suggestions with compromising side-effects are given only half a score.

Perception of Application and Explanations. For each scenario, we asked participants their perception of Laκsa's **Behavior Appropriateness** (7-point Likert scale) and if they agreed or disagreed that the explanations were helpful (**Explanation Helpfulness**; 7-point).

4 Results

We recruited 18 participants (11 females) with ages 19 to 65 (Median=26) years. 9 participants were graduate students, and three were undergraduates. P01, P16, and P17 were students in a computer-related field. P18 was a web programmer, while the others spanned a wide range of areas (e.g., actor, pianist, field interviewer, hospital administrator, chemical engineering, retiree). We engaged each participant for 1h 44min on average (range: 1h 29m to 1h 58m). Each participant was compensated $20.

Although participants experienced the same scenarios, due to conducting the experiment in the field, there was some variability in what Laκsa sensed and the resulting explanations. For example, location accuracy depended on where the participant walked to, weather, and other environmental factors; when the participant walks to the café in S4, she may hear background music, or be near people who are talking.

For S4, participants exhibited two distinct behaviors to explore the hypothetical situation: (i) they either just sat where they were and tried to use the What If explanation facility (S4-if, 10 cases), or (ii) walked to the café to test Laκsa *in-situ* (S4-situ, 11 cases; some participants did both). We treat these as distinct scenarios.

Perception of Application Behavior and Explanations. As expected, participants perceived Laκsa's behavior as inappropriate for S2 and S3 ($M_{S2}=-2.1$, $M_{S3}=-2.4$), but appropriate for S4 ($M_{S4\text{-}if}=2.0$, $M_{S4\text{-}situ}=2.4$): $F_{3,25}=3.90$, $p<.05$; contrast test: $p<.01$. Participants generally found the explanations helpful (M=1.5), though explanations were less helpful in S2 ($M_{S2}=0.6$) than in S4 ($M_{S4\text{-}if}=2.4$, $M_{S4\text{-}situ}=2.5$); Tukey HSD test: $p<.05$.

Intelligibility Usage. Combining usage logs across S2 to S4, we determined participants' overall usage of intelligibility (see Table 1), and their usage for each explanation type (see Table 2.). Most participants actively looked at many Explanation Types (Median=8), many times (View Count Median=21), for about 3 minutes per scenario. They also tended to look more at deeper contexts (Place or Sound) than just Availability (Context Ratio Median=1.4). Usage ranged from very engaged (View Count Max=65, Scenario Duration Max=12.5min), to conservative, e.g., min 2 views (P08S4-if), 1 explanation type (P14S4-situ), scenario duration <1 min (P08S4-if).

Table 2. illustrates which explanation types were more popular, i.e., higher view count, and how much time participants spent looking at each explanation type.

Table 1. Summary statistics of intelligibility usage per participant scenario (S2-S4).

Per Scenario	Mean	SD	Std. Err.	Min	Median	Max
View Count	24	15	2	2	21	65
# Explanation Types	7.9	3.4	0.4	1	8	19
Context Ratio	1.8	1.8	0.2	0	1.4	8
Total Duration (s)	205	136	18	52	196	749

Table 2. Usage of explanation types: total view count of explanation types for all participant scenarios, and median durations for respective views (for Total View Count > 15).

	Total View Count			Median Duration (s)		
	Avail.	Place	Sound	Avail.	Place	Sound
What + Certainty	232	114	69	5.7	3.1	3.2
History	130	5	3	.5	-	-
Outputs + Certainty	84	102	33	6.4	5.6	4.9
Inputs	217	45	60	7.1	6.4	6.0
Why	26	16	44	3.5	9.9	6.1
Why Alt	21	35	41	4.7	9.6	4.9
What If	31	-	-	24.8	-	-
Definition	5	7	28	-	-	6.1
Situation	-	-	15			

User Understanding and Control Suggestions. For each scenario, participants articulated 0 to 8 correct beliefs about Laksa's behavior (Median=4). 41% of the beliefs were about the awareness of the inferred Value for Place and Sound, 28% about a broader understanding of the inference (Alternative Values and Certainty), 15% about the Inputs state and Model mechanism, and 2.5% about deeper Technical details. 14% of the beliefs were drawn from the Situation to justify Laksa's behavior.

Participants provided 0 to 6 correct Control Suggestions (Median=2) for each scenario, and had an average Control Score of 2.10 (Std Err=0.29). This is significantly greater than 1 (i.e., H_0: Score>1, p<.01). Participants made effective and partial Control Suggestions about: Availability Rules (29%), Settings (27% Place, 8% Sound), Behavior Change (36%).

The extent and pattern of intelligibility usage did affect how well participants understood Laksa, as we shall see next.

Impact of Intelligibility Usage on Understanding. We chose View Count and Context Ratio as factors of intelligibility usage. We split View Count into discrete intervals of 10 counts; we split Context Ratio into two groups Shallower (N=34) and Deeper (N=20), where participants saw twice as many explanations about Place or Sound than Availability (ratio ≥2). Fig. 2 summarize these results showing that higher and deeper use of explanations lead to higher Understanding and Control scores.

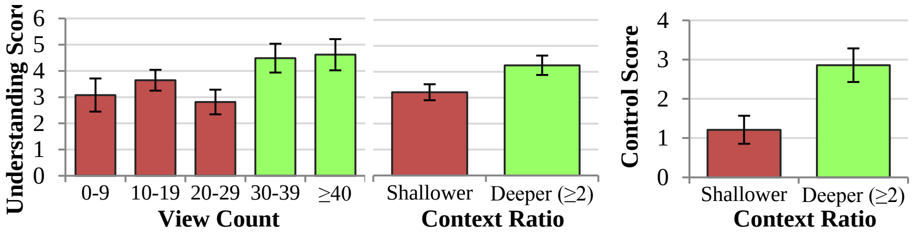


Fig. 2. (Left) Understanding Score is higher when explanation views ≥30 than less (p<.05) and **(Right)** Understanding and Control Scores are higher (both p<.05) when participants ask more explanations about Deeper contexts (Place and Sound) than Availability.

5 Discussion and Recommendations

Our results show how participants were willing to use intelligibility, and how quickly or deeply they used it. This satisfies our hypothesis that more Intelligibility Usage (View Count and Context Ratio) improves Understanding. These have implications on how intelligibility should be provided to facilitate its more effective use.

5.1 Usage and Usefulness of Intelligibility

Our results show that intelligibility was *useful* for participants to (i) engage with intelligibility (some participants deeply so), (ii) rate explanations as helpful, and (iii) better understand application behavior. We next discuss how they used intelligibility, and how certain usage patterns were more effective in improving user understanding.

Diverse Usage of Explanation Types. Participants used a diverse range of explanation types and in diverse ways. What and Inputs were conduits to other explanations for participants to learn deeper reasons. However, although some explanation types were used less than others, some were viewed for longer durations (*e.g.*, Place Why / Alt). Furthermore, as with [7], the sequence diagrams of our participants revealed various usage styles (e.g., quick comparison between Why and Why Alt reasons, diving into a deeper context after going straight to Availability Inputs).

Unlike what was found in [4], our participants felt that the What If explanation was easy to use and liked it (e.g., P11S1: *"[Using] it was just more fun … I like to think of hypothetical things, but it also gives me a sense of what the phone is capable of, and helps to develop trust when you know what to expect"*). In fact, for S4, 10 participants chose to ask What If instead of immediately walking to the café. However, this fascination with What If can also give users false trust since it obscures potential pitfalls in sensing. Participants who used What If in S4 may not realize how noisy the café may be or that the Place inference was not particularly good there. P11 did not bother to explore Laκsa's inference in-situ because *"technology is supposed to make your life easier; you shouldn't have to waste time to make sure it works right."* Perhaps providing warnings that sensing can fluctuate due to environmental conditions may help users be more careful when using What If.

Occasionally, participants forgot what had recently happened, e.g., for S2, P07 thought he was talking to the experimenter at the time Laksa inferred Sound as Talking. Had he played the recorded audio of that time (Situation), he would have learned that only singing was heard. Using the played audio, P15 and P16 were able to identify guitar sounds when Sound was finally recognized correctly as Music. Hence, in combination with History, Situation explanations can help jog a user's memory of what was happening, independent of the application's inference. This helps them form Situation Justifications for the application behavior. How may we also provide Situation explanations for contexts other than Sound? For Place, perhaps by showing a photograph at the location (if one was taken at the same time). For Motion recognition, perhaps by animating an *interpreted* diagram of how the phone was moving (derived from accelerometer data).

While earlier research into intelligibility sought to prioritize providing some explanation types over others (e.g., [4, 5]), along with [8], our findings suggest instead to provide a diversity of explanation types will be helpful to support different learning and troubleshooting strategies users have.

Deeper Usage of Intelligibility. Our quantitative results indicate that viewing more explanations, especially about deeper contexts can lead to deeper understanding, and more effective control suggestions for improving the application behavior. So, to promote user understanding, we need to encourage users to dig for more explanations, and to dig deeper. Perhaps, if the user starts asking questions, the application can hypothesize faults, and highlight which factors are probably causing them. These guesses could come from a knowledge base of typical faults [7], or be triggered when inferences Certainty becomes too low (e.g., <80% [8]).

5.2 Constraints for Intelligibility

While the upper bounds of our participants' usage of intelligibility may give an indication of engagement, the lower bound may portend the limits to which some users are willing to use intelligibility. Therefore, we derive some time and view constraints for intelligibility. Participants only spent about 3-10 seconds viewing each explanation, so each explanation page needs to be correctly and effectively interpreted within that short duration. Perhaps, if an explanation cannot be understood within that duration, it should be split into multiple parts where the user can ask for more *on demand*. Furthermore, our quicker participants spared only about 1-3 minutes exploring explanations for each incident. This may be even shorter without the experimenter demand effect when users explore intelligibility outside of a user study. Hence, question asking should be streamlined to facilitate multiple views (~20) within about 2 minutes before the user gives up. In our scenarios, we have focused on investigating the usage of intelligibility about incidences *in-situ* and in the moment. However, users may postpone investigating an incident until they have more time. Under those circumstances, we expect usage amounts and duration to be higher.

6 Conclusion

We have presented a quasi-field study measuring how participants used an intelligible context-aware application in scenarios representing real-world, "everyday" situations. We found that viewing more explanations, especially more about deeper contexts can further improve user understanding of application inference. We provided implications for promoting more effective intelligibility usage, time constraints within which users are willing to view intelligibility, and discuss how much intelligibility should be provided to sufficiently improve user understanding of context-aware applications.

Acknowledgements. This work was funded by the National Science Foundation under grant 0746428, Intel Research, and A*STAR, Singapore.

REFERENCES

1. Cheverst, K. *et al.* (2005). Exploring issues of user model transparency and proactive behavior in an office environment control system. *UMUAI 05*, 15(3-4), 235-273.
2. Dey, A.K., Abowd, G.D. & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *HCI Journal*, 16(2–4): 97–166.
3. Hilbert, D.M., & Redmiles, D.F. (2000). Extracting usability information from user interface events. ACM *Computing Survey* 32(4), 384-421.
4. Lim, B.Y. *et al*. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. *CHI 09*, 2119-2128.
5. Lim, B.Y., & Dey, A.K. (2009). Assessing Demand for Intelligibility in Context-Aware Applications. *Ubicomp 09*, 195-204.
6. Lim, B.Y., & Dey, A.K. (2010). Toolkit to Support Intelligibility in Context-Aware Applications. *Ubicomp 10*, 13-22.
7. Lim, B.Y., & Dey, A.K. (2011). Design of an Intelligible Mobile Context-Aware Application. *MobileHCI 11*, to appear.
8. Lim, B.Y., & Dey, A.K. (2011). Investigating Intelligibility for Uncertain Context-Aware Applications. *Ubicomp 11*, to appear.
9. Lu, H. *et al*. (2009). SoundSense: scalable sound sensing for people-centric applications on mobile phones. *MobiSys 09*, 165-178.
10. Milewski, A.E., & Smith, T.M. (2000). Providing Presence Cues to Telephone Users. *CSCW 00*, 89-96.
11. Muir, B. (1994). Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37(11): 1905–1922.
12. Rosenthal, S., Dey, A.K., Veloso, M. (2011). Using Decision-Theoretic Experience Sampling to Build Personalized Mobile Phone Interruption Models. *Pervasive 11*, 170-187.
13. Roto, V. *et al.* (2004). Examining Mobile Phone Use in the Wild with Quasi-Experimentation. Helsinky Institute for Information Technology (HIIT), Technical Report.
14. Tullio, J. *et al.* (2007). How it works: A field study of non-technical users interacting with an intelligent system. *CHI 07*, 31-40.
15. Vermeulen, J. *et al*. (2010). PervasiveCrystal: Asking and Answering Why and Why Not Questions about Pervasive Computing Applications. *IE 10*, 271-276.
16. Weka for Android. https://github.com/rjmarsan/Weka-for-Android. Retrieved 26th August 2011.