

Investigating Intelligibility for Uncertain Context-Aware Applications

Brian Y. Lim, Anind K. Dey
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
{byl, anind}@cs.cmu.edu

ABSTRACT

Context-aware applications use sensing and inference to attempt to determine users' contexts, and take appropriate action. However, they are prone to uncertainty, and this may compromise the trust users have in them. Providing intelligibility has been proposed to help explain to users how context-aware applications work in order to improve user impressions of them. However, we hypothesize that intelligibility may actually be harmful for applications that are very uncertain of their actions. We conducted a large controlled study of a location-aware and a sound-aware application, investigating the impact of intelligibility on understanding, and user impression of applications with varying certainty. We found that intelligibility impacts user impressions, depending on the application's certainty and behavior appropriateness. Intelligibility is helpful for applications with high certainty, but it is harmful if applications behave appropriately, yet display low certainty.

Author Keywords

Context-awareness, intelligibility, explanations, user study.

ACM Classification Keywords

H.m. Information systems: Miscellaneous.

General Terms

Human Factors, Reliability.

INTRODUCTION

Context-aware applications make use of sensing and intelligent inference to automatically learn users' contexts and adapt their behavior [7]. However, much of the context sensing is done invisibly [21], and the context inferencing is growing increasingly complex (*e.g.*, numerous rules, hidden Markov models). Lay users may not understand how these applications make their decisions, let alone be aware when decisions are made and actions are taken. This can lead to user frustration, and loss of trust in the applications [18]. Therefore, context-aware applications should be *intelligible*

(also called transparent, comprehensible, scrutible), capable of generating explanations of their behavior [3]. Towards this endeavor, much research has shown the positive effects of providing users with explanations from applications. In several domains including decision-making [9], end-user debugging [12], and user modeling [5], explanations have increased user trust and acceptance of applications. In context-aware systems, Lim *et al.* [14] found that some explanation types were more effective than others in improving understanding and trust, and later investigated more explanation types that end-users of context-aware applications are interested in [15].

Even though these studies show great promise for the efficacy of intelligibility in context-aware applications, they have assumed the use of systems that have reasonably high certainty in their actions, and that, while fallible, generally take appropriate actions. Intelligibility would enhance the positive impression a user may have of an application, and reveal how it intelligently tries to figure out what is happening even for difficult sensing and inference situations. Unfortunately, because of these difficulties in sensing and inference, applications can be uncertain of their actions, often resulting in users having a negative impression of these applications. It is hoped that intelligibility would help bring up this shortfall, and raise a user's impression of a context-aware application. However, is there a certainty below which intelligibility would not help, but may actually *harm* a user's impression of the application? If this were the case, the user could lose even more trust in the application's capability and precision. So an application with sufficiently low certainty would not benefit from adding intelligibility, and instead, the developer should focus on improving its certainty instead.

In this paper, we present two scenario-driven lab studies where we investigate the interaction between intelligibility and application uncertainty. For the first study, we manipulated the provision of Intelligibility in three levels (None, Certainty-only, Full), and Certainty in six levels (50, 60, 70, 80, 90, 100%), in a *between-subject* design for an online survey. We designed two context-aware applications (location-aware, and sound-aware) to explore the impact of certainty on intelligibility for applications with differing complexity. In a follow-up study, we ran a think-aloud study using a reduced form of the online survey, seeking to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UbiComp '11, September 17–21, 2011, Beijing, China.

Copyright 2011 ACM 978-1-4503-0630-0/11/09...\$10.00.

add greater context to our quantitative findings. Our contributions are:

1. Understanding how users respond to intelligibility in context-aware applications under different levels of certainty; and
2. Identifying when, how, and why intelligibility is helpful or harmful as a result of application certainty.

INTELLIGIBILITY AND UNCERTAINTY

In this section, we provide background and related work on intelligibility, uncertainty in context-aware applications, and the impact of showing uncertainty to end-users.

Displaying Uncertainty in Context-Aware Applications

Context-aware applications are prone to uncertainty, and one common strategy for dealing with this involves user mediation where the user resolves uncertainty [8]. Furthermore, these applications should represent to their users what they know [3], not hide this ambiguity or uncertainty [10], and reveal the "seams" of their underlying systems [4]. Currently, some context-aware applications are able to model uncertainty due to their underlying probabilistic models (e.g., [12, 20]), but few display the system certainty (e.g., [5]).

Conversely, many studies have also explored various ways to display uncertainty, and the benefits of doing so. Antifakos and colleagues showed that uncertainty improved task performance speed of participants when certainty is high [1], and that participants verified automatic settings made by a context-aware system less often when its certainty was high or medium [2]. Similarly, Rukzio *et al.* found that displaying uncertainty slowed down user performance, because users would double-check fields with lower certainty [19]. In studies of presenting location information, visualizations of location certainty were found to improve user performance with location-based services [6, 13]. Though not explicitly investigating about uncertainty, Yan *et al.* found that displaying higher trust and reputation values of mobile applications increased users' willingness to continue using them [22].

Our work adds to the research on displaying uncertainty by carefully varying uncertainty to identify a certainty threshold below which displaying uncertainty becomes harmful instead of helpful, in two different contexts — location and sound. Furthermore, we extend the displaying of uncertainty to include other explanations that provide users with a fuller form of intelligibility.

Intelligibility in Context-Aware Applications

For this work, we use the definition of intelligibility defined by Lim *et al.* [14, 15], which classifies explanations in terms of questions that users may ask of context-aware applications. Specifically, we developed interfaces for explanations of the following questions:

1. **What** is the current value of the context?
2. **Certainty**: how certain is the application of this value?
3. **Why** is this context the current value?

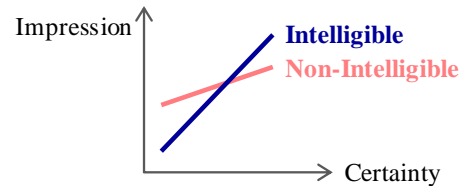


Figure 1. Hypothesis 1: Intelligibility will improve user impressions when an application is certain of its actions, but it will harm impressions when it is uncertain. Only interaction effect suggested, not linearity of trends.

4. **Why Not**: why isn't this context value Y, instead?
5. **Inputs**: what factors affect this context?

We describe how to provide explanations for these questions in a later section (Application Platforms).

HYPOTHESES

While we do not assert it here, we believe the user's impression of an application impacts her trust of it. We define that a user has a good *impression* of a context-aware application when she perceives it to be highly **certain** of its inference, feels that it generally behaves **appropriately**, and she **agrees** with what it is doing. As illustrated in Figure 1, we hypothesize that:

H1a: Above a certainty threshold, intelligibility improves a user's impression of a context-aware application.

H1b: Below the threshold, intelligibility harms the user's impression of the application. This could be due to the user realizing how poorly the application is performing.

We hypothesize that this effect on impression is due to the increased understanding provided by intelligibility:

H2: Providing intelligibility helps increase a user's understanding of the application.

While H2 has been shown to be true in [14], we seek to verify those results, as H1 depends on this. Thus, H2 in combination with H1b hypothesizes that a *gain* of understanding about a low certainty application leads to a *loss* in impression. Next, we describe a large-scale, between-subjects lab study to test these hypotheses.

METHOD

We are primarily interested in the interaction between the provision of intelligibility, the certainty of the application, and the impact on understanding and impression. We chose to investigate this effect using a large-scale, controlled lab study. The study was deployed online through Amazon Mechanical Turk (MTurk) to allow us to collect input from a large number of participants and span many levels of certainty and intelligibility (as in [14, 15]). For generality, we designed two context-aware applications and varied their certainty, and intelligibility levels. We exposed participants to several canonical situations of these applications through 10 different scenarios.

Experimental Conditions

We varied Intelligibility and application Certainty as independent variables in a *between-subject* experiment, across two applications, for a total of $3 \times 6 \times 2 = 36$ conditions.

Intelligibility (3 conditions: None, Certainty-only, Full)

We varied whether participants were provided with explanations where they only saw the application inference (None), or additionally saw a rich explanation visualization (Full). We included an intermediate intelligibility level, where we provided just Certainty percentage only, to investigate how much value the explanation visualizations add over just showing certainty.

Certainty (6 conditions: 50%, 60%, 70%, 80%, 90%, 100%)

We varied certainty as six intervals (rather than a dichotomy) to be able to observe any trends that may arise.

Measures

We are interested in measuring how much participants understand the application for each intelligibility condition, and whether this affects their perception of certainty, feeling of whether the application behaved appropriately, and how much they agree with the application's inference.

Understanding. For each scenario, we asked participants *why* the application inferred what it did, and *why not* something else (free-text). We asked these questions for all scenarios to prime participants to think about the underlying inference of the application. We analyzed the responses from the sixth of 10 scenarios presented, as we expected participants to be sufficiently familiarized with the application through previous scenarios, but not overly tired of providing feedback. We validated that this was true through a sampling of the responses.

Perceived Certainty. For each scenario, we asked participants how certain they believed the application was in its inference (as numerical input 0 to 100%). After the scenarios, we asked for their overall sense of the certainty.

Perceived Appropriateness. For each scenario, we measured what the participant felt about the appropriateness of the application behavior, on a 7-point Likert scale from Very Inappropriate to Very Appropriate.

Agreement. For each scenario, we measured how much the participant agreed with the application's inference, given the ease or difficulty of making the inference; on a 7-point Likert scale from Strongly Disagree to Strongly Agree.

APPLICATION PLATFORMS

To investigate the interaction between intelligibility and uncertainty, we designed two applications — LocateMe and HearMe — and varied their certainty and intelligibility levels. Derived from design explorations of intelligibility in [16], both applications are mobile phone applications, but deal with different contexts (location, and sound activity, respectively), different inference mechanisms, and different explanation interfaces. While real, physical prototypes were not used in this study, these applications have been prototyped, and their described functionality are feasible and indicative of real applications and their associated uncertainty. We describe these applications, how they sense and make inferences, their basis for uncertainty, and how they visualize their inferences.

Each application has three different levels of intelligibility. The None version would just show the output of the application (e.g., "You are at the Washroom", "You were in a Conversation"). The Certainty version adds a certainty percentage (e.g., 89%, 62%). The Full version adds an explanation visualization (see Table 1 and Table 2).

LocateMe

LocateMe is a location-aware mobile phone application that uses GPS, Wi-Fi and cellular networks to triangulate where the user is, and match that to a predetermined set of locations to infer which *place* the user is at. It then uses this inference to take actions such as sending a reminder, or identifying the nearest printer. In the scenarios, LocateMe is used for indoor and outdoor situations.

Basis for uncertainty. Due to the probabilistic model of the user's location (as a Gaussian area), LocateMe infers the user being at places with varying levels of certainty. Its certainty depends on how much the user's estimated area "overlaps" with the area of the named place, and is computed into a probability. The larger the area of the named place, and/or the closer the user's area is to that place, the higher the certainty. Uncertainty is also affected by sensing errors due to GPS signal occlusion (e.g., being indoors), Wi-Fi or Cell network signal strength, etc.

HearMe

HearMe is a sound-aware mobile phone application that uses the phone's microphone to sense and infer one of three activities: whether the user is (i) in a conversation, (ii) listening to music, or there is mostly (iii) ambient noise. It uses several features extracted from processing the microphone signal, such as: frequency bandwidth, spectral entropy, low-energy frame rate, Mel-Frequency Cepstral Coefficients (see [16] for more details about features used). HearMe uses a trained naïve Bayes model to infer whether the sound heard was one of the three activities.

Basis for uncertainty. HearMe models uncertainty of its inference from the probabilistic uncertainty of the naïve Bayes model. This depends on the sound samples used to train the original model. HearMe does not model the error due to the microphone signal for uncertainty.

Due to the ubiquity of GPS devices and location sensing in smart phones, LocateMe is likely more familiar to users than HearMe which uses machine learning inferences that are less common-place in devices available to consumers.

SCENARIOS

Similar to [1, 14, 15], we use scenarios to let participants learn about and experience our applications. However, rather than present 5-second video clips to help participants experience a scenario, we provided users with a precise representation to understand the ground truth of each scenario. For LocateMe, we showed a map or floorplan indicating where the participant would actually be in the scenario. For HearMe, we played an audio clip of what the participant and her phone would supposedly have heard.



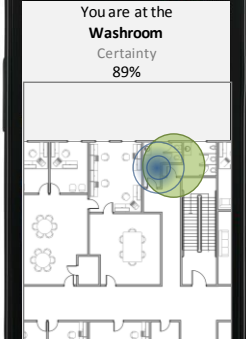
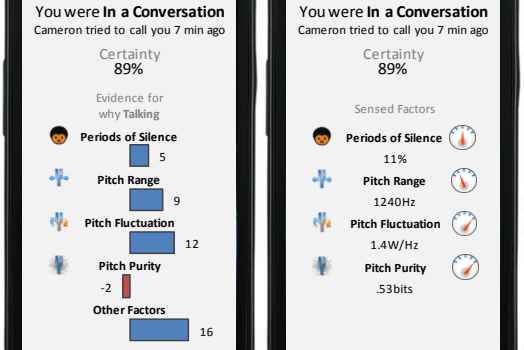
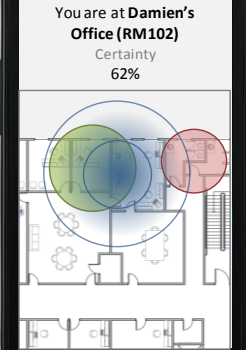
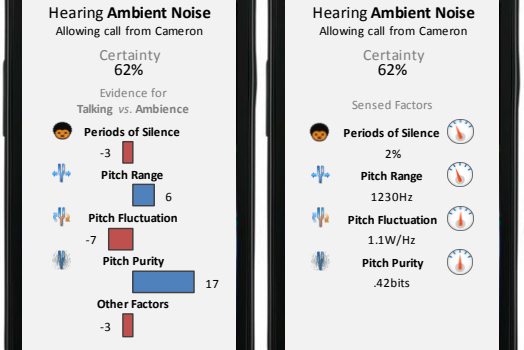
	LocateMe	HearMe
Situation description and Ground truth	<p>(a) You are in the washroom taking care of some business, just before your meeting with your neighboring coworker, Damien.</p>  <p>Star denotes where you actually are at; purple triangle denotes Damien's office (RM102).</p>	<p>(d) At the coffee shop, you find Michelle, a coworker, there, and have a chat with her.</p> <p><i>Participant listens to an auto-started audio clip of ambient noise in a coffee shop, with a female voice occasionally talking.</i></p> <p>H6-groundtruth.mp3</p> 
Application behavior (Certainty 90%)	<p>(b) You receive a text message from Damien, who tells you he is waiting for you at his office. You check LocateMe:</p> 	<p>(e) You are not interrupted for 12 min, and when the conversation ends, you receive a notification message from HearMe:</p>  <p>You see that Cameron had tried to call you, but HearMe suppressed his call since it interpreted you as uninterruptible.</p>
(Certainty 60%) Wrong inference, in this case	<p>(c) Damien calls to ask where you are since LocateMe said you are in his office, which is obviously false. You check LocateMe:</p> 	<p>(f) You are not interrupted for 12 min, and when the conversation ends, you receive a notification message from HearMe:</p> 
Explanation UI Description	<p>LocateMe uses "bubbles", to determine and show where the user is, rather than showing pin-point positions. The user's sensed location is represented by two blue concentric circles, and a Gaussian blue area. He is most likely to be in the center of the area, but less likely the further away from it. The bounds indicate thresholds of certainty (50, 90%) that the user is within the bounds. Places are represented with uniform circular areas of varying size. <i>E.g.</i>, Washroom is defined with a circle, with the center where the room is, and the size is how large the room is.</p> <p>The user is inferred to be at a place if his blue bubble "overlaps" with the place's bubble. A green bubble indicates the place where the user is inferred to be; a red bubble indicates where he is not inferred to be. (b) shows a green bubble over the washroom overlapping with the user's blue bubble to explain <i>why</i> he is inferred to be there. The large overlap suggests a high certainty (in this case, 89%). (c) explains <i>why</i> the user is <i>not</i> inferred to be at the washroom but at Damien's office instead, by showing: a red bubble for the washroom, a green bubble for Damien's Office. The blue bubble overlaps with the green bubble more than with the red bubble, indicating 62% certainty.</p>	<p>HearMe uses two types of visualizations to explain what it senses, and how it infers an activity, with a Sensed State and Evidence visualization, respectively. We substitute the technical names of the input factors with metaphorical terms (<i>e.g.</i>, Periods of Silence for low-energy frame rate, Pitch Purity for spectral entropy), and aggregate the remaining factors as Other Factors. We explain the meanings of each factor and implications of their values, <i>e.g.</i>: Periods of Silence indicates what percentage of the sound sample was relatively silent compared to the rest of it; talking would have higher percentage. The Sensed Factors viz (right diagram) shows the values of the factors, and a gauge icon indicating whether each value is at, below, or above the average values for that factor.</p> <p>The Evidence viz (left diagram) shows a bar chart indicating if each factor votes for (blue towards right) or against (red towards left) the inference, and by how much. This viz can be used to compare one output against all others (e), or specifically contrast between two outcomes (f). The balance of the bars indicate how certain the application is about its inference. If it is more certain, the bars are weighted more towards the right, and if less certain, the bars are equally weighted to the right and left.</p>

Table 1. Scenario scripts, application interfaces showing Full intelligibility, and their interpretation of Scenario 6.

	Description / Function	LocateMe	HearMe
What	Show the current inference of the context and consequent action.	Reports inferred place, and shows blue bubble of the user in map visualization.	Reports inferred sound activity.
Certainty	Show the certainty of the application's inference of the current context value.	Shows a certainty percentage, and size of bubbles in map visualization (larger sizes show lower certainty).	Shows a certainty percentage, and sense of balance of bars in evidence viz (more balance show lower certainty).
Why	Show a model-based explanation of how the application inferred the current context value.	Shows the overlap between the inferred place (as a green bubble) and the user's blue bubble.	Shows evidences for the inference due to each input factor in a bar chart visualization.
Why Not	Show a model-based explanation distinguishing how the application did not infer the alternative inference.	In addition to the Why visualization, shows the lack of overlap between the place (as a red bubble) and blue bubble.	Shows the evidence visualization, contrasting the current inference against the alternative inference.
Inputs	Show the current values of input context / features.	Visually shows the user's position by positioning the blue bubble in a map.	Lists current input factor values, and provides a gauge of its relative value.

Table 2. Explanation types. LocateMe uses a map and bubbles visualization for its explanations about its location inference.

HearMe uses lists the current values of its sensed factors, and their corresponding evidence to explain its sound activity inference.

We presented 10 scenarios as a chronological sequence of events happening through a single day. As in [15], the scenarios were written to span five *themes* typical of what context-aware applications are used for: interruption management, social awareness, reminders, recommender, exploration / learning. Each theme is *repeated* twice (not consecutively) to provide repeated exposure. Collectively, the scenarios are *representative* of the application certainty (e.g., for 60%, the application behaved appropriately for 6 out of 10 scenarios). Hence, participants in the None intelligibility condition could perceive the certainty of the application. The certainties presented (for Certainty and Full intelligibility) also reflected the certainty condition, but with small *randomized* differences (e.g., 60, 63, 59, 60, 58, 62, 61, 57, 60, 60%), to prevent participants from ignoring the values had they been constantly shown 60% repeatedly.

Table 1 shows the scripts and diagrams shown to participants in the LocateMe (left), and HearMe (right) surveys for a scenario, S6. Next, we describe what participants were asked to do for each application survey.

PROCEDURE

After consenting to participate in the survey (either LocateMe or HearMe), the participant was randomly assigned to a Certainty condition and an Intelligibility condition. He read instructions on how the application works, and how to interpret its display. As recommended by [10], we then asked two verification questions (multiple-choice) to ensure comprehension. The participant next went through 10 scenarios to experience the application under various situations. For each scenario, he read (i) a scenario description, and (ii) the subsequent response of the application which may or may not be appropriate for the situation. He was then (iii) asked verification questions to ensure he had carefully read and understood the scenario. Next we asked questions for our measures of (iv) perception of certainty, (v) application behavior appropriateness, and (vi) agreement. Finally, he was asked about his (vii) understanding of the application inference. After the scenarios, he was asked about his background with using smart phones, and for demographic information.

PARTICIPANTS AND DATA CLEANSING

We recruited participants from Amazon Mechanical Turk. There were 397 incomplete HITs (human intelligence tasks), and 584 completed HITs. We rejected 76 HITs because each participant had low verification score, rushed through the survey too quickly, and/or was unconscientious (gave reasons that were gibberish, repetitive, or irrelevant). Of the remaining 508 participants, their survey completion time was Median=33 minutes (8.9 to 109), and their verification score was Median=20 (7 to 22) out of 22. Some participants had low verification scores, which indicates poor understanding of the scenarios and application, but their free-text reasons indicated conscientious effort in the survey. So they were included in our population sample to represent users who have greater comprehension difficulty. We had participants across 36 conditions (3 Intelligibility × 6 Certainty × 2 Application) in our experiment (M=14.1, 11 to 17 in each condition). We paid each participant \$2.

DATA ANALYSIS AND RESULTS

In this section, we present the analysis we performed on the survey results, related to our hypotheses. Before we investigate whether intelligibility influences users' impressions of a context-aware application, first we analyze whether intelligibility improves understanding of how the application works (H2). We assume that understanding is not influenced by the certainty of the application.

Understanding of Application Inference

As a measure of understanding, we coded the free-text responses about how they thought the application made its inferences for S6. We counted how many of the reasons *about the application* that they provided were correct. A reason is considered correct if it relates to an actual factor that the application uses (e.g., GPS, latitude, distance threshold, bubbles; Periods of Silence, Pitch Purity, noisiness). We eliminated repeated and redundant reasons

Intelligibility	None	Certainty	Full
LocateMe	.73 ± .10	.81 ± .10	.97 ± .09*
HearMe	.72 ± .08	.82 ± .08	1.57 ± .09**

Table 3. Number of correct reasons (Mean ± Standard Error) counted from participant free-text reasons of how the application made its inference in S6. Contrast of None vs. Full: p<.01 for HearMe, p=.08 for LocateMe*.**

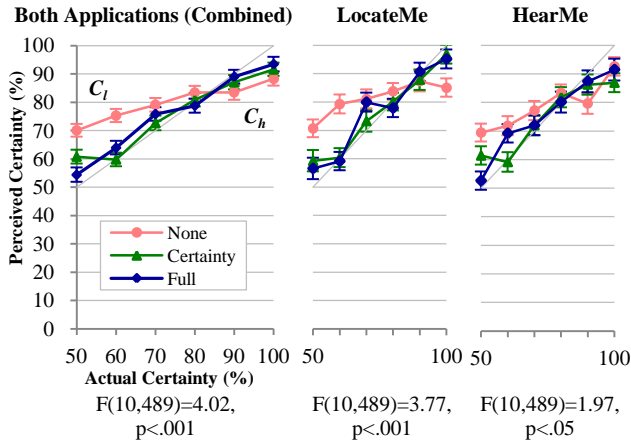


Figure 2. Perception of Overall Certainty: combined analysis (Left), and for individual applications (Middle and Right). Participants with Full intelligibility perceived a higher certainty when the application had high actual certainty, but perceived a lower certainty when it had low actual certainty.

App	Combined		LocateMe		HearMe	
Actual Certainty	Low 50-70%	High 90-100%	Low 50-60%	High 100%	Low 50-70%	High 80-100%
None vs. Full	p<.001	p<.05	p<.001	p<.05	p<.01	p=n.s.
None vs. Certainty	p<.001	p=n.s.	p<.001	p<.05	p<.01	p=n.s.

Table 4. Pre-hoc contrast between Intelligibility types for low and high actual certainty. These groups were chosen after visually inspecting the interaction graphs.

Certainty (%)	50	60	70	80	90	100
Certainty	<.01	n.s.	n.s.	n.s.	.05	.01
Full	.02	n.s.	<.01	n.s.	n.s.	<.01

Table 5. Means testing of whether perceptions of overall certainty are different from actual certainties. *t*-test *p*-values suggest copying if *p*=n.s.

(*i.e.*, paraphrasing of the same idea), and accepted language that demonstrated an approximate idea of valid concepts.

We fit a mixed model with: correct reason count as the dependent variable, intelligibility and application as independent variables, interaction \times application as an interaction effect, certainty as a control variable, and participant as a random variable (nested in intelligibility, application, and certainty). We found that participants with Full intelligibility gave more correct reasons than those with None, especially regarding HearMe (see Table 3).

Next, we analyze whether and how this increase in understanding influences how participants perceived the certainty of the application. We annotate some figures to indicate notable findings in our results (*e.g.*, C_l , $C_{-a,l}$, A_a).

Perceived Overall Certainty

We asked each participant about their perception of the overall (average) certainty of the application, after completing all 10 scenarios. To examine differences in this perception for each application separately, we fit a mixed model with: perceived overall certainty as dependent variable, intelligibility and certainty as independent variables, intelligibility \times certainty as an interaction effect,

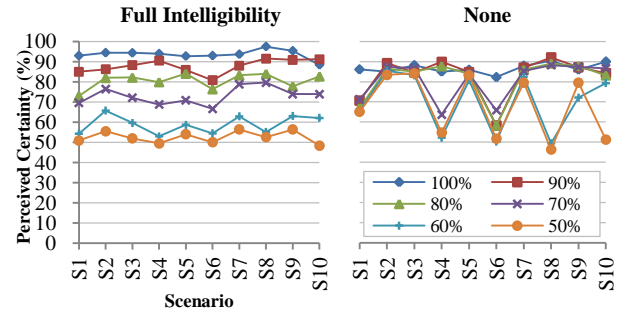


Figure 3. Perceived certainty influenced by application appropriateness across scenarios. The application behaved appropriately for at least one Certainty condition in S1, S4, S6, S8, and S10, more so for lower certainty conditions.

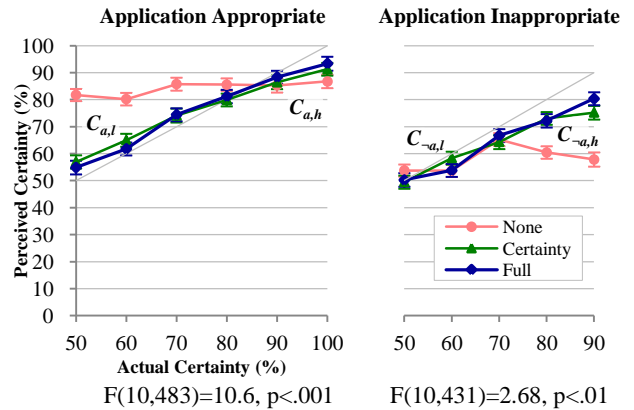


Figure 4. Perceived certainty across actual certainty by application appropriateness. Note: no inappropriate scenarios for 100% certainty condition.

App Behavior	Appropriate		Inappropriate	
Actual Certainty	50-70%	80-90%	50-70%	80-90%
None vs. Full	p<.001	p<.05	p=n.s.	p<.001
None vs. Certainty	p<.001	p=n.s.	p=n.s.	p<.001

Table 6. Contrast between Intelligibility types for low and high actual certainty grouped by application behavior.

and participant as a random variable (nested in intelligibility and certainty). We also combined data from both applications, and fit a similar mixed model but also with application as a control variable. These results are presented in Figure 2 and Table 4. Our results show that, for high actual certainty, participants with intelligibility perceived a higher certainty than those without (C_h); for low actual certainty, participants with intelligibility perceived a lower certainty than those without (C_l). Alternatively, an interpretation may be participants with intelligibility just copied the certainty displayed. Means testing suggests that this could be so (see Table 5), but we further investigate this in a follow-up study (see later).

While our results show that participants' perceived overall certainty *across* different certainty levels is influenced by intelligibility, we next show that their perception also varies based on how the application behaved per scenario.

Perceived Certainty by Application Appropriateness

To investigate perception of certainty across scenarios, we analyzed the repeated measure of how certain participants

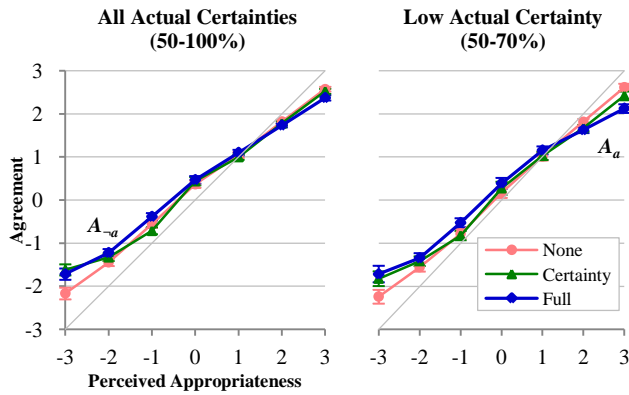


Figure 5. Agreement across Perceived Appropriateness, grouped by actual certainty. The effect of *finding* 4_h is only significant for low actual certainty.

Actual Certainty	All (50-100%)		Low (50-70%)	
Appropriateness	Not (<0)	High (2-3)	Not (<0)	High (2-3)
None vs. Full	p<.01	p<.05	p<.01	p<.01
None vs. Certainty	p=.052	p=n.s.	p=n.s.	p<.05

Table 7. Contrast between Intelligibility types for low and high appropriateness grouped by actual certainty.

felt the application was for each scenario. Figure 3 (Right) shows the fluctuation of perceived certainty as participants with no intelligibility (None) go through the scenarios, depending on whether the application behaved appropriately in the scenario. When participants received Full intelligibility (Figure 3, Left), their perceived certainty was more stratified, and less fluctuating.

We group our results by application appropriateness, and fit two mixed models with: perceived certainty as dependent variable, intelligibility and certainty as independent variables, intelligibility \times certainty as an interaction effect, and participant as a random variable (nested in intelligibility and certainty). Our results (see Figure 4 and Figure 5) show that, for appropriate application behaviors, participants with intelligibility perceived lower certainty than those with None when encountering actual low certainty ($C_{a,l}$), and conversely perceived higher certainty when encountering actual high certainty ($C_{a,h}$). For inappropriate application behaviors, there was no difference in perception for actual low certainty ($C_{-a,l}$), but participants with intelligibility perceived higher certainty for actual high certainty, than participants without ($C_{-a,h}$).

Agreement by Perceived Appropriateness

Given the difference in perception due to application appropriateness, we are interested to see if participants' opinion of how appropriately the application behaved, and how much they agree with its inference were affected by intelligibility. These self-reported, repeated measures for every scenario were obtained with the following questions:

Perceived Appropriateness with "How appropriately or inappropriately did the application behave in this situation?" (7-point Likert scale)

Agreement with "How much do you agree or disagree with the application's inference, given how easy or difficult it is to infer this?" (7-point Likert scale)

We did not compare perceived certainty, because, as expected, it varied independently of appropriateness.

We fit a mixed model with: agreement as dependent variable, appropriateness and agreement as independent variables, appropriateness \times agreement as an interaction effect, and participant as a random variable (nested in appropriateness and agreement). Our results (see Figure 5 and Table 7) show that participants tended to agree with the application when they perceived it behaved appropriately, and *vice versa*. When participants felt the application behaved inappropriately (<0), those with intelligibility agreed with the application more than those with None (A_{-a}). However, when participants perceived the application behaved very appropriately (2-3), participants using an application with low certainty and intelligibility agreed less with it than those with None (Figure 5, Right; A_a).

Summary of Findings

We summarize our findings in terms of our hypotheses. Participants with Full intelligibility gave more correct reasons of how the application works, than those without (satisfies **H2**). *Finding* C_h satisfies **H1a** that intelligibility improves user impression of a context-aware application if its certainty is high. This is more pronounced when the application behaved inappropriately ($C_{-a,h}$) than appropriately ($C_{a,h}$). Conversely, *finding* C_l satisfies **H1b** that intelligibility harms user impression if its certainty is low; particularly, when the application behaved appropriately ($C_{a,l}$), or when it displays a low certainty despite being appropriate (A_a). However, participants with intelligibility disagreed less with the application inference when they felt that it behaved inappropriately (A_{-a}). To gain better insights into our results, we ran a follow-up study where we engaged participants face-to-face.

FOLLOW-UP: THINK-ALOUD STUDY

At this point, our results positively support our hypotheses that intelligibility *exaggerates* the perception of certainty compared to not receiving any explanation. However, this could be because our participants with intelligibility could just be copying the certainty value they were shown (see Table 5). Do participants mindlessly copy these values, or do they weigh their opinion with previous experiences with the application (from previous scenarios)? Furthermore, *finding* A_l suggests that Full intelligibility provides some additional benefit of improving the perception of certainty than showing Certainty-only. How does Full intelligibility help to reinforce the Certainty information provided?

Method and Procedure

Answering these questions will help us examine the link between the information provided through Certainty-only and Full intelligibility, the user's understanding, and their subsequent impression of the application. It will also help us explore whether and how Full intelligibility influences

Certainty	Low (S6:52%, S9:49%)				High (S6:89%, S9:92%)			
Application	LocateMe		HearMe		LocateMe		HearMe	
Participant	1	5	2	8	3	5	4	7

Table 8. Distribution of participants in think-aloud study. Each participant saw S6 (appropriate behavior) and S9 (inappropriate), iterated within-subjects with intelligibility types in the order: None, Certainty, Full.

the user compared to Certainty-only. To explore this, we ran a follow-up *think-aloud* study where we presented all three intelligibility conditions *within-subject*, focusing on a subset of Certainty conditions (low: 50%; high 90%). We continued to use both applications (*between-subject*) due to their differences in complexity, and participant reliance on their explanations. Hence, we have four conditions. Due to the time-consuming nature of the think-aloud study (one-hour long), we presented only two scenarios (S6, S9) to our participants, counter-balanced for application correctness. S6 has been described in Table 1. For S9, LocateMe correctly infers the user in Meeting Room B and automatically loads the meeting agenda; HearMe correctly infers conversation during a group meeting, and allows the user to retrieve the audio and save it. In the follow-up study, the application always behaves incorrectly for S6, but correctly for S9, regardless of certainty.

We recruited two participants per condition (total 8), 4 females, mean age 28.1 years old (21 to 58). We presented three iterations of the survey starting with None, Certainty, then Full, so as to avoid a training effect. Each scenario has the same format and questions as the original survey. Additionally, we asked them to think-aloud and provide reasons for their answers. This way, we learned about how they thought the application made its inferences, and how they constructed opinions of the application's behavior. We used paper surveys, so participants could refer to previous surveys, compare previous phone displays, their previous answers, and discuss why they changed or did not change their opinions. Table 8 shows which conditions participants P1 to P8 were in. We discuss our findings in the next section in the context of our original quantitative results.

DISCUSSION

We discuss the results from both experiments in terms of how intelligibility affects understanding (H2), and how it affects users' impression of context-aware applications (H1).

H2: Intelligibility Increases Understanding of Context-Aware Applications

As expected, Full intelligibility allowed participants to better express an understanding of the applications. This was particularly significant for HearMe, because the explanations listed relevant factors, increasing the participants' vocabulary to describe how the application works. In the think-aloud study, participants could analyze and interpret the values of HearMe's sensed factors, and their corresponding evidences, and LocateMe's bubble visualization. However, because the input factors were not explicitly stated in LocateMe as they were for HearMe, participants gave reasons for how LocateMe works by

describing names of technologies, *e.g.*, GPS, "position-specific sensing" (P3), a "grid in the building" (P6), or in terms of the situation, *e.g.*, signal blocked by nearby stairs (P5), or improved signal because of proximity to windows (P5). Full intelligibility only marginally increased the correct ideas that participants had about how LocateMe works. For HearMe, without Full intelligibility, participants considered the "noisiness" of the audio, along with speaker identification (especially that of the user) as the most important factors for inference, but with Full intelligibility, they tended to discard their original understanding and described the inference in terms of the factors shown. We can interpret the differences between the applications as due to their *complexity* and the users' *familiarity* with them. These findings reinforce those in [14] that prior knowledge about an application domain (in this case, LBS) reduces the impact of intelligibility on understanding.

By providing more information, intelligibility also helps provide participants with an increased awareness of what the application was inferring, and what it understood. This consequently impacted their impression of it.

H1: Impact of Intelligibility on User Impressions

Without Intelligibility, MTurk participants are influenced by whether the application behaved appropriately to perceive its certainty (see Figure 3, Right). They perceived a modestly high certainty (~85%) when it is behaved appropriately, and how often it behaved appropriately (Figure 2), but perceived a low certainty otherwise (~60%). Their overall perceived certainty is also impacted by the cumulative application behavior, gently increasing from ~70 to ~90% as actual certainty increases from 50 to 100% (Figure 3). With intelligibility, participants' perceived certainty aligned more closely with the actual certainty.

But, do participants just copy the application certainty (as suggested in Table 5)? In the think-aloud study, though influenced by the displayed value, *all* participants did not outright adhere to it. They continued to be influenced by their perception of how difficult it was to make the inference, and whether the application behaved appropriately, but adjusted their certainty rating depending on the presented value. Hence, if a low certainty was presented, participants lowered their certainty estimate, and while if a high certainty was displayed, participants raised their certainty estimate, but not all the way to the presented value for both cases. Furthermore, participants reevaluated their certainty rating when given Full intelligibility. Next, we discuss and interpret our results (shown in Figure 4 and Figure 5) in terms of hypotheses H1a and H1b. We found a caveat to H1b which we denote as H1b'. Table 9 summarizes these positive and negative impacts that intelligibility has on user impressions.

H1a: Intelligibility Increases User Impressions of Context-Aware Applications with High Certainty

For context-aware applications with **high certainty**, our results verify previous findings of [14, 15] that

intelligibility improves users' impression of the applications (*finding C_h*). With intelligibility, participants perceived a higher certainty from the application, particularly when it **behaved appropriately** (*finding C_{a,h}*). After seeing the Certainty-only intelligibility, P4 raised her original rating (85% with None) to "just below" what was shown (92%), despite feeling that HearMe was "overconfident," because of her "overall understanding" of the conversation that she heard in the audio clip. With Full intelligibility, participants felt the explanations "reinforced" the high certainty (P3), or even raised their certainty of the application (P6).

Intelligibility had a more significant impact on perceived certainty if the application **behaved inappropriately**, since MTurk participants had a lower baseline certainty rating (about 60% instead of ~85%). Though not to as high a level for appropriate application behavior, intelligibility raised their confidence rating by a larger margin (by 15% to ~75%; *finding C_{-a,h}*). With Certainty-only intelligibility, P3 liked that LocateMe was "honest," and trusted it more. P4 felt that HearMe "would know its own certainty better than [she] would" and raised her certainty to 75-85%, which was between what she had imagined and what was presented. P6 insisted that LocateMe's certainty should not have been so high, but raised her rating by 5%. With Full intelligibility, participants reevaluated their opinion. P4 and P7 were more accepting of HearMe's certainty, and raised their ratings.

H1b: Intelligibility Decreases Impressions of Applications with Low Certainty when they behave Appropriately

For context-aware applications with **low certainty**, intelligibility revealed how uncertain they were, and compromised the impressions participants had of them (*finding C_l*). This was particularly notable when the application **behaved appropriately** (*finding C_{a,l}*). In the think-aloud study, participants were surprised to discover the low certainty. For LocateMe, P5 thought that the location in S9 was easier to infer than in S6, and felt that the certainty should have been higher (60%) than the presented 49%. For HearMe, P2 felt that the conversation in S9 was "so clear" that the certainty should be higher at 60-65%. Furthermore, the unexpectedly low presented certainty caused participants to disagree more with the application inference (*finding A_n*). P1 and P5 lowered their agreement rating from 7 (None) to 2 (Certainty), and P8 from 6 to 4. With Full intelligibility, P8 became convinced by HearMe's displayed 50% certainty by examining the bar chart of evidences and noting they were very balanced; she consequently lowered her certainty rating.

H1b': Intelligibility Increases Impressions of Applications with Low Certainty when they behave Inappropriately

Contrary to H1b, intelligibility was helpful for an application with **low certainty** when it **behaved inappropriately** (*finding A_a*), even though it did not influence perceived certainty (*finding C_{-a,l}*). Participants appreciated the difficulty of inference, forgave the application, and disagreed less with it (*A_l*). P5 conceded that it was "very difficult" for LocateMe to estimate the

		Certainty	
		Low	High
Behavior Appropriateness	Overall	Harmful (H1b) Decreases perceived overall accuracy (<i>finding C_l</i>).	Helpful (H1a) Increases perceived overall accuracy (<i>finding C_h</i>).
	Appropriate	Harmful (H1b) Decreases perceived accuracy (<i>finding C_{a,l}</i>), and Decreases agreement with inference (<i>finding A_n</i>).	Helpful (H1a) Increases perceived accuracy (<i>finding C_{a,h}</i>).
	Not Appropriate	Helpful (H1b') Increases agreement with inference (<i>finding A_{-a}</i>).	Helpful (H1a) Increases perceived accuracy (<i>finding C_{-a,h}</i>).

Table 9. Impact of Intelligibility on user impressions of a context-aware application depends on application certainty and whether it behaved appropriately.

certainty, and thought "it got it almost right"; she agreed more with the application, changing her rating from 4 (None) to 5 (Certainty). With Full intelligibility, participants more clearly saw how uncertain the applications were: large margins of error (LocateMe), or a high amount of ambiguity (HearMe). This allowed P8 to understand how HearMe "misjudged the environment," and "agree with its logic based on the parameters."

DESIGN RECOMMENDATIONS

There are two ways to apply our findings in terms of application certainty: regarding *overall* certainty, or *per situation* certainty. Considering overall certainty, our findings recommend providing intelligibility as long as the application usually has high certainty. However, our findings caution that intelligibility is harmful if certainty is too low, so intelligibility should not be provided for such applications; their certainty should be improved first. While the precise threshold for what is a *sufficiently* high overall certainty depends on the application and domain, our results (see Figure 2) suggest it falls within the range of about 80-90% for a non-critical, "everyday" application.

Considering certainty per situation, we note that an application that usually has high certainty may still occasionally have situations with low certainty. Fortunately, in the majority of times with high certainty, we still recommend providing intelligibility even if it is ultimately wrong and behaves inappropriately. On the other hand, our recommendation is not immediately clear with low certainty. The impact of intelligibility on impression also depends on whether the application behaves appropriately.

If the application behaves appropriately, showing intelligibility compromises the original good impression the user may have had, causing her to lower her impression. If it behaves inappropriately, intelligibility can help her realize how difficult the inference task is, and improve her impression. Unfortunately, an application will not be able to know if it will act appropriately beforehand. It will be safer to not show intelligibility before it acts, when certainty is low. After it acts, if the user asks questions, especially if a

why not question, it is likely the application behaved inappropriately, where it is beneficial to show intelligibility. Therefore, just show intelligibility *on demand*, when situation certainty is low. Our results (see Figure 4, Left) indicate that our participants have a baseline belief that the application is about 80-85% certainty when it behaved appropriately. This suggests that, for each situation, a context-aware application may safely provide intelligibility automatically when it is at least 80% certain, but should provide intelligibility on demand when it is less certain.

Carefully designing explanations can provide an alternative solution to deal with intelligibility as a double-edged sword for low certainty. Intelligibility should focus on convincing the user how difficult the inference task is, and how the application is intelligently tackling it, rather than implying that the application is incompetent. This could mean not revealing the low certainty in explanations. For example, HearMe's Sensed Factors visualization explains what *input values* it knows, but does not betray HearMe's uncertainty.

CONCLUSION AND FUTURE WORK

We have described a large controlled study investigating the impact of intelligibility on understanding and user impressions of context-aware applications with varying certainty from low to high. This was conducted using lab-based, scenario-driven surveys of two context-aware applications (location-aware, and sound-aware). Our results show that intelligibility can positive or negatively impact user impressions, depending on the application's certainty and behavior appropriateness. Intelligibility is helpful for applications with high certainty, but it is harmful for applications with low certainty, because the user loses even more trust in its capability. Still, intelligibility can help users appreciate and forgive applications if they behave inappropriately and have low certainty. This work explicitly cautions the necessity for a context-aware application to be sufficiently certain before it leverages intelligibility.

In this study, we have focused on a *passive* display for intelligibility, and did not have users act on the information; they could only use intelligibility to judge their impression of the application. However, users could also *interactively* use intelligibility for debugging and finding out why the application faltered (*e.g.* [12]). Perhaps, this could make intelligibility useful instead of harmful to user impression.

This work provides a stepping stone to understanding how intelligibility affects a user's impression of a context-aware application. For future work, we plan to gain more lucid and nuanced insights into the use of intelligibility and how that affects users in real-world situations through deploying an intelligible prototype in a longitudinal field trial.

ACKNOWLEDGMENTS

This work was funded by the National Science Foundation under grant 0746428, and the Agency for Science Technology And Research, Singapore. We also thank Matt L. Lee, Tawanna Dillahunt, Denzil Ferreira, and Jin-Hyuk

Hong for their helpful discussions; Gabriela Marcu for her involvement in the think-aloud study; and our reviewers for their insightful feedback.

REFERENCES

1. Antifakos, S. *et al.* (2004). Evaluating the Effects of Displaying Uncertainty in Context-Aware Applications. *UbiComp 04*, 54-69.
2. Antifakos, S. *et al.* (2005). Towards improving trust in context-aware systems by displaying system confidence. *MobileHCI 05*, 9-14.
3. Bellotti, V. & Edwards, W.K. (2001). Intelligibility and Accountability: Human Considerations in Context-Aware Systems, *Human-Computer Interaction*, 16(2-4): 193-212.
4. Chalmers, M. & MacColl, I. (2003). Seamless and seamful design in ubiquitous computing. In *Workshop: At the Crossroads: The Interaction of HCI and Systems Issues*, *UbiComp 03*.
5. 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.
6. Dearman, D. *et al.* (2007). An exploration of location error estimation. *UbiComp 07*, 181-198.
7. 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.
8. Dey, A. *et al.* (2002). Distributed mediation of ambiguous context in aware environments. *UIST 02*, 121-130.
9. Dzindolet, M. *et al.* (2003). The role of trust in automation reliance, *International Journal of Human-Computer Studies*, 58(6): 697-718.
10. Greenberg, S. (2001). Context as a dynamic construct. *HCI 16(2)*, 257-268.
11. Kittur, A. *et al.* Crowdsourcing user studies with Mechanical Turk. *CHI 08*, 453-456.
12. Kulesza, T. *et al.* (2009). Fixing the Program My Computer Learned: Barriers for End-users, Challenges for the Machine. *IUI 09*, 187-196.
13. Lemelson, H., *et al.* (2008). A Study on User Acceptance of Error Visualization Techniques. *HUCUBIS 08*.
14. Lim, B.Y. *et al.* (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. *CHI 09*, 2119-2128.
15. Lim, B.Y., Dey, A.K. (2009). Assessing Demand for Intelligibility in Context-Aware Applications. *UbiComp 09*, 195-204.
16. Lim, B.Y., Dey, A.K. (2011). Design of an Intelligible Mobile Context-Aware Application. To appear in *MobileHCI 11*.
17. Lu, H. *et al.* (2009). SoundSense: scalable sound sensing for people-centric applications on mobile phones. *MobiSys 09*, 165-178.
18. 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.
19. Rukzio, E. *et al.* (2006). Visualization of uncertainty in context aware mobile applications. *MobileHCI 06*, 247-250.
20. Tullio, J. *et al.* (2007). How it works: A field study of non-technical users interacting with an intelligent system. *CHI 07*, 31-40.
21. Weiser, M. & Brown, J.S. (1997). The coming age of calm technology. *Beyond Calculation: the Next Fifty Years*, 75-85.
22. Yan, Z. *et al.* (2010). Effects of displaying trust information on mobile application usage. *ATC 10*, 107-121.