Design of an Intelligible Mobile Context-Aware Application

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 are increasingly complex and autonomous, and research has indicated that explanations can help users better understand and ultimately trust their autonomous behavior. However, it is still unclear how to effectively *present* and *provide* these explanations. This work builds on previous work to make context-aware applications intelligible by supporting a suite of explanations using eight question types (*e.g.*, Why, Why Not, What If). We present a formative study on design and usability issues for making an *intelligible* real-world, mobile context-aware application, focusing on the use of intelligibility for the mobile contexts of availability, place, motion, and sound activity. We discuss design strategies that we considered, findings of explanation use, and design recommendations to make intelligibility more usable.

Author Keywords
Context-awareness, intelligibility, explanations, user study.
ACM Classification Keywords
H.m. Information systems: Miscellaneous.
General Terms
Design, Human Factors.

INTRODUCTION

Context-aware applications make use of sensed inputs and contexts, coupled with intelligent decision-making to automatically and calmly adapt to serve users [4]. However, much of the context sensing is done invisibly [19], and context inference is growing increasingly complex (using large rule-sets, hidden Markov models, *etc.*). 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 [14]. To counter this, context-aware applications should be *intelligible* (also called transparent, comprehensible, scrutable) by providing explanations of their behavior [1, 5, 10, 11].

Indeed, there have already been several context-aware applications that support some level of intelligibility. Cheverst *et al.*'s Intelligent Office System [2] exposes sensor values, and explains its fuzzy decision tree model that controls office appliances. Tullio *et al.*'s interruptibility displays [16] explain how they determine a manager's interruptibility by exposing the values of sensors in the manager's room. Kuleza *et al.* built an email sorting application [7] that supports Why (*i.e.*, why the application took a particular action) and Why Not (*i.e.*, why the application did not take a different action) explanations for its naïve Bayes classifier. Vermeulen *et al.*'s PervasiveCrystal [17] also supports Why and Why Not explanations but for ambient environments. These systems support a limited set of explanations users can ask for: What, Input values, Why, and Why Not. However, Lim & Dey [10] found that users ask a wider range of questions of context-aware applications, and that different explanations have different impacts on user understanding [9].

To support this wider range of explanations for context-aware applications, the Intelligibility Toolkit [11] was developed to automatically generate eight question type explanations: What, Why, Why Not, How To, What If, Inputs, Outputs, and Certainty. While this significantly helps facilitate the development of intelligible context-aware applications, it is unclear how to effectively *present* these explanations. We advance the knowledge of how to design for intelligibility by investigating explanation design for a real-world, mobile context-aware prototype. In this work, we focused on intelligibility for the mobile contexts of availability, place, motion, and sound activity.

Our contributions are the:

1. Exploration of *design* and *usability* issues in making a context-aware application intelligible, and
2. Provision of design recommendations to address them.

The rest of the paper is organized as follows: we describe the prototype that we developed to be intelligible. We then describe our rationale for designing the explanations to make them more interpretable. To discover how users use intelligibility and the usability issues that they face, we ran an *in-situ*, scenario-driven, think-aloud study. We describe our experimental set up, method and data analysis. We then describe how our participants used explanations, and discuss our interpretations of some factors influencing their behavior. We observed how participants used explanations differently depending on their goals, and how some participants encountered difficulties due to their lack of prior knowledge or their chosen problem solving strategies. Finally, we provide design recommendations arising from these observations, and describe our plans for future work.

MobileHCI 2011, Aug 30–Sept 2, 2011, Stockholm, Sweden.
Copyright 2011 ACM 978-1-4503-0541-9/11/08-09....$10.00.

LAKSA — SOCIAL AWARENESS APPLICATION

Staying aware of others is an established need that people have [15]. Between close friends and family members, this can help people feel a greater sense of connectedness as each person goes about their daily activities. Between coworkers, this can inform people of the most suitable times to contact them. We have developed Laκsa, a *mobile* application that shares people's availability status. Laκsa is an acronym for Location, Activity, Connectivity (κ), and Social Awareness that describe its function. Availability is determined from six lower-level contexts: Place, Motion, Sound activity, phone Ringer, Schedule, and who is enquiring (Contactor). While similar to CenseMe [13], it uses a slightly different set of contexts, aggregates them into an availability context through rules (rather than just showing all contexts as *presence* information), and is intelligible, such that it can explain its complex behavior.

Our focus in this paper is the use of Laκsa as a vehicle to explore the design, implementation, and use of intelligibility in a sophisticated, multi-factor context-aware application. In the rest of this section, we shall describe the various contexts that Laκsa employs, briefly describing their implementation. Laκsa is designed to support the complexities of availability in social relations by considering availability as *multi-faceted*, and *multi-factored*. One's availability depends on who is asking (different *facet* for different viewer), and on contextual *factors* (*e.g.*, Place, Motion, Sound). Figure 1 describes Laκsa's context hierarchy. We designed Laκsa to be sophisticated in using many contexts, and complex sensing and inference mechanisms, to exemplify how context-aware applications can manage many factors that users would find cognitively difficult. We anticipate users will ask for explanations to determine or remember Laκsa's complex mechanisms. Next we describe the contexts Laκsa models.

Availability: *Available, Semi-Available, Unavailable* — is determined based on rules regarding the following six factors. The contactor interprets the availability and decides whether to contact and how (call, text, email, *etc.*).

Contactor (Who is Enquiring): *Family, Friend, Coworker,* and *Default* (to check user's default status) — categorizes person contacting or enquiring about the user.

Place: *Home, Office, Café, Library, etc.* — represents the semantic location of the user. It is computed by sensing latitude and longitude from the Skyhook Wi-Fi API (uses a hybrid GPS, Wi-Fi, and cell tower positioning algorithm), and matching to a pre-determined named location that the user specifies. To convey accuracy, it also reports the sensed distance error and detected number of access points.

Motion: *Sitting, Walking, Cycling, Placing the phone Flat, etc.* — represents the user's physical activity inferred with the phone placed in a front pants pocket. Inferences are made with a decision tree trained using activities from several users. Features extracted from the accelerometer are similar to [8]: *e.g.*, mean and standard deviation for three axes, phone orientation angles, and signal powers.

Top-tier context	Availability	Intelligibility Explanations
Lower-tier contexts	Place, Motion, Sound, Ringer, Schedule, Contactor	
Lower-tier input features	Latitude, Longitude, Energy, Mean Standard Deviation, MFCCs, *etc.*	
Sensors and Sources	GPS, Wi-Fi, Accelerometer, Microphone, Phone State, Calendar, *etc.*	▨

Figure 1. Laκsa context architecture with different tiers of context used to infer higher-level tiers. The user sees the Availability status, and the intelligibility explanations.

Sound: *Talking, Music, and Ambient Noise* — represents the sound activity that Laκsa recognizes from what it can hear from the phone's microphone. Inferences come from a naïve Bayes classifier trained on sound samples. Features extracted are similar to [12]: *e.g.*, mean and standard deviation of power, low-energy frame rate, spectral flux, spectral entropy, spectral centroid, bandwidth, Mel-Frequency Cepstral Coefficients (MFCCs).

Phone Ringer: *Silent, Vibration, or Normal.*

Calendar Schedule: *Personal, Work, or Unscheduled.*

While one could compare the use of explanations for different contexts (*e.g.*, Place, Motion, Sound) and different decision models (rules, decision trees, naïve Bayes), for this formative work, we focus on exploring the design and use of explanations across this breadth of factors.

DESIGN OF INTELLIGIBILITY

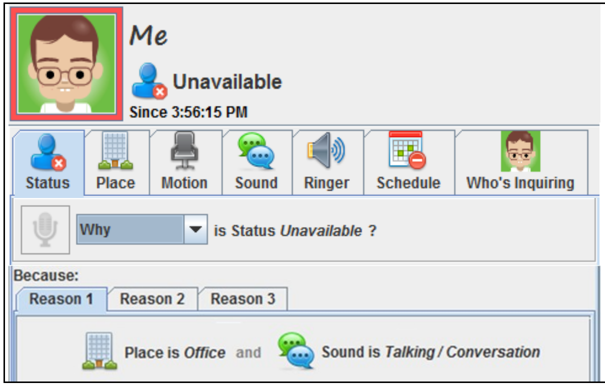
Having defined the context types, we seek to make Laκsa intelligible so that users can understand what it knows and how it makes decisions about user availability. We employ explanations supported by the Intelligibility Toolkit [11]:

1. **What** is the value of the context?
2. **Why** is this context inferred as the current value?
3. **Why Not**: why isn't this context inferred as Y, instead?
4. **How To**: when would this context take value Y?
5. **Inputs**: what details affect this context? (Factors, input features, related details, *etc.*)
6. **Outputs**: what other values can this context take?
7. **What if** the conditions are different, what would this context be? (Requires user manipulation)
8. **Certainty**: how confident is Laκsa of this value?
9. **Description**: meaning of the context terms and values.

Explanations generated from the Intelligibility Toolkit contain the information content to answer these nine questions and can be rendered using simple text templates. However, these may not be easy for lay users to interpret or quickly assimilate. Therefore, we employed several design strategies to help make the explanations more usable. Figure 2 and Figure 3 show some explanation UIs resulting from the strategies described next.

Reducing and Aggregating Explanations

We expect that users of context-aware applications would rather be focused on their day-to-day tasks than dedicate too much attention to technical details. Hence, it is important to simplify and *reduce* the provided explanations to be concise and salient. We have done this for Laκsa by aggregating explanation types (*e.g.*, What value and



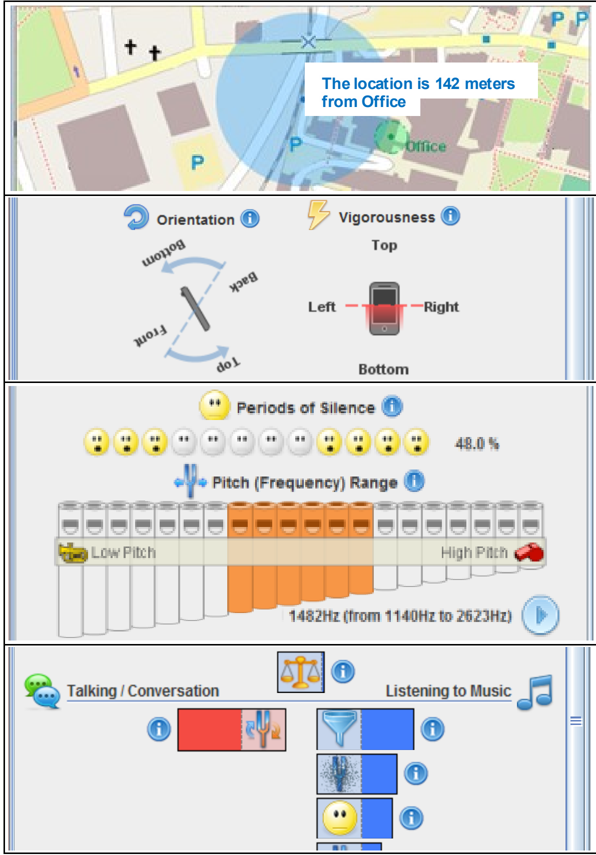
(a) This is the core information to show the Availability status that Laksa has inferred. It shows the user or buddy's picture, name, and Availability status. It also shows the time since the last change.

(b) To ask for an explanation, first select which context you would like to ask about by selecting one of these seven Context Tabs. Each tab shows the current value of the context (What explanation), *e.g.*, Sound = Talking.

(c) Next, you ask a specific question from the drop-down menu. The Question Panel adapts to the question selected (*e.g.*, Why, Why Not, What If).

(d) The resulting explanation is rendered in the Explanation Panel. **List** visualization: the left screenshot shows a reason with a list of multiple factors. Some explanations have multiple alternative reasons, shown in different sub-tabs.

Figure 2. Screenshot of the Laksa showing how to use the components in the core and intelligibility user interface.



(a) Map visualization explaining with 'bubble' components. Blue bubble indicates estimate of current location; green dotted bubble indicates specified region of named location.

Why is my Place inferred as "Office"? Because your estimated location bubble overlaps with the bubble specifying where Office is.

(b) Physical Diagram visualization illustrating the interpreted mechanical motion of the phone leading to its inference of cycling (two shown here). Dotted lines and arrows or shading show boundary conditions of rules of the decision tree model. Users can mouse over the diagram to see textual explanations with numerical values.

Why isn't Motion inferred as "Cycling"? Because the phone is oriented more than the drawn angle, the vigorousness detected is below the illustrated value, *etc*.

(c) Metaphorical visualization of sound feature values that was sensed and computed at a specified moment (two shown here). This can show *current* values sensed, or *average* values for each possible outcome.

What details (Inputs) affected the Sound inference? 48% of the sound heard was relatively silent (Periods of Silence); the range of pitches heard was from 1140 to 2623Hz (this can be played aurally for the user to listen to), *etc*.

(d) Weights of Evidence visualization**.** Bar chart visualization showing weights of evidence voting for or against the inference of Talking. Weights are represented by the length of each bar; color and positioning indicate evidence for or against. Not all factors shown.

Why isn't Sound inferred as "Talking"? Because most of the factors vote for the inference of Music (especially hidden features, the sound Volume, Periods of Silence, *etc*); only Pitch Fluctuation votes for Talking.

Figure 3. Explanation Visualizations rendered in the explanation panel. Some examples and their interpretations. The UI uses icons derived from http://www.fatcow.com/free-icons.

Certainty rating shown together), reducing the number of reasons, length of reasons, and number of input features (*e.g.*, omitting MFCCs for sound); and combining explanations for simultaneous consumption (*e.g.*, presenting x-y-z accelerometer values in 2D diagrams). While this may compromise comprehensiveness, it is intended to make the explanations more interpretable.

Visualizing Explanations

Users should more quickly assimilate visual explanations because of the higher bandwidth of diagrams [18]. Hence, we provide several visual representations: icons for context and feature values, dynamic diagrams that change when values change, and even animation and sound (to hear pitches and pitch ranges). Visualizations are customized for the context domains (*e.g.*, map for Place, physical 2D phone diagram for Motion). Since Sound is not visual, we chose to explain sound by metaphor (*e.g.*, showing a pan flute to represent pitches and ranges; see Figure 3c).

Explaining in Simple and Relatable Terms

As we deploy intelligibility in real-world prototypes, users need to understand how the contexts and recognition features relate to the real world, using lay concepts. Therefore, we simplify names (*e.g.*, "spectral entropy" renamed to "pitch pureness", "low-energy frame rate" renamed to "periods of silence"), normalize numerical values to lay scales (*e.g.*, 0 to 100), and include Description

	Description / Function	**Availability**	**Place**	**Motion**	**Sound**
What	Shows current value / state of context.	Value	Map (user's location bubble)	Value	Value
Why	Shows a model-based explanation of how Laκsa inferred the current output value (outcome).	List (multiple conditions)	Map (user & actual place bubbles *overlapping*)	Physical Diagram (with boundary conditions)	Weights of Evidence
Why Not	Shows a model-based explanation distinguishing how Laκsa did not infer the alternative output value.	*Multiple* Lists (multiple conditions)	Map (user & desired place bubbles *separated*)	Physical Diagram (with boundary conditions)	Weights of Evidence
How To	Shows a model-based explanation of how Laκsa typically or generally infers the target output value.	List (required conditions)	Map (actual place bubble)	Physical Diagram (of *average* input values for desired outcome)	Metaphorical Viz (of *average* input values for desired outcome)
Inputs	Shows the current values of input context / features.	List (current input values)	List (latitude, longitude)	Physical Diagram (of *current* input values)	Metaphorical Viz (of *current* input values)
Outputs	Shows possible output values the context may take.	List (*possible* output values)			
What If	Shows a UI to allow the user to specify different input values and see what the output value would be.	Editable List (of current input values)			
Certainty	Shows the certainty or confidence of Laκsa 's inference of the current context value.		List (distance error, number of access points)	% Certainty (of inference)	% Certainty (of inference)
Desc.	Shows terminology or description of context / feature.	Text (sentences)			

Table 1. Explanation Visualization Types for each context, and question type. Only the What value is shown for Ringer, Schedule, and Contactor. Note that Certainty is shown together with both What and Outputs, *i.e.* users can see the certainty the current and alternative outcomes, respectively. Visualization types described in Figure 2 and Figure 3.

explanations that describe what each context factor and feature mean. Descriptions are presented in physical rather than system terms (*e.g.*, there are more *periods of silence* for talking than ambient noise, because there are significant pauses in speech that are relatively quiet). Furthermore, features for Motion and Sound were scaled to physically meaningful names and values (*e.g.*, *vigorousness* to represent accelerometer signal power in *Watts*). We also chose to visualize explanations for motion using *first principles*. For example, orientation information is shown as the orientation of the phone relative to the ground (see Figure 3b). Note that ensuring that terms are domain relatable may require significant domain knowledge, rather than just naïvely applying effective features identified in the literature about activity recognition (*e.g.*, [8, 12]).

Providing Explanations for Control

We chose to provide explanation types for each context only if users could leverage the information to improve Laκsa, or change their behavior. What If explanations are only provided for the top-tier Availability context. For other contexts, users would not be able to meaningfully change the input features (*e.g.*, changing the entropy or frequency to influence sound). Furthermore, we omit trivial explanations, *e.g.*, asking What If one is at a specific coordinate location to learn which semantic place Laκsa would infer the user being at; asking Why Not questions about manually set contexts (*e.g.*, schedule or ringer mode).

We iterated on the design of Laκsa with these strategies and feedback from colleagues who are active HCI researchers.

LAKSA PROTOTYPE IMPLEMENTATION

We built Laκsa using a client-server architecture. Sensing of low-level contexts (*e.g.*, latitude, longitude coordinates, accelerometer, microphone) was performed on an Android mobile phone, and some intermediate features (*e.g.*, discrete Fourier transform, entropy, energy) were computed on the phone. The extracted features are then sent via XMPP to a server for further processing. On the server, we modeled the contexts for users with the Enactor framework [5], and used the Intelligibility Toolkit [11] to support the *querying* for various questions, *generation* and *reduction* of explanations about the contexts, and *presentation* of the explanations in various graphical and textual formats.

To measure how participants use the Laκsa interface, and to support rapid prototyping, we developed the interface on a touch screen tablet, rather than on the mobile phone. Users interact with the UI with a mouse, pen stylus, or finger. The latter interaction closely resembles cell phone interaction.

LAKSA PROTOTYPE USAGE

Users ask for explanations by selecting a question from the drop down menu (see Figure 2a). Each question only pertains to the particular context of focus (*e.g.*, Availability). To ask about another context, the user selects the context by its tab (Figure 2b), and asks the questions in the panel (Figure 2c). The resulting explanation appears in the Explanation Panel (Figure 2d; Figure 3; Table 1).

SCENARIO-DRIVEN THINK ALOUD USER STUDY

To explore the use of the intelligibility features in Laκsa, we conducted a *scenario*-driven user study where participants *think aloud* as they used it. This was an exploratory study where we investigated how and why participants used intelligibility, and how this use impacts their understanding of Laκsa. We conducted an *in-situ* controlled study rather than a field deployment for two main reasons: (i) to present participants with a lower-fidelity interface to elicit more feedback from the think aloud study, and (ii) to avoid having serious usability issues that could confound results in a field study, where it would also be harder to monitor participants' usage and rationale.

Procedure

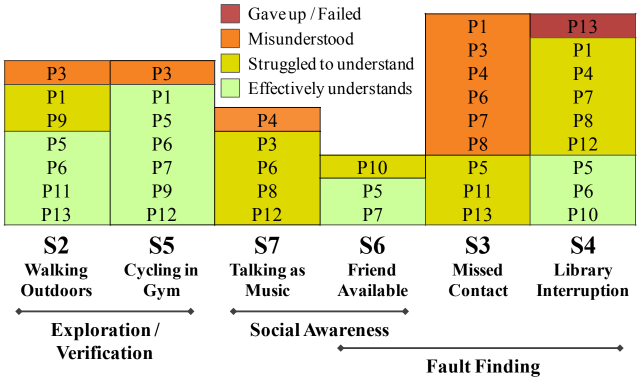
The experiment began with the first scenario (see later) as a training session, where the experimenter explained the function of the Laκsa prototype as an availability awareness application, its sensing capabilities, and its intelligibility features. Participants verified that they understood the interface and explanations by stepping through the interface themselves and thinking aloud what they understood Each subsequent scenario involved participants moving around the university campus to a respective location (*e.g.*, library, café, office) and engaging in a specific activity (*e.g.*, walking, cycling). The experimenter shadowed the participant all the while. The participants were then told of an *incident* (*e.g.*, having your phone ring loudly while searching for a book in the library) and the phone's subsequent behavior. They were asked for their opinion of the situation, and of the behavior of the application. They were then asked to explore Laκsa to find out what is really happening or to clarify the situation. The participants were prompted to think aloud as they did this, and the experimenter probed for clarification. For each scenario, we recorded what participants did and said during the think aloud. Finally, we interviewed them about their opinion and understanding of how Laκsa sensed and inferred contexts.

Controlled In-Situ Scenarios

The user study was scenario-driven to expose participants to a wide range of situations they may encounter with Laκsa. To increase the visceral quality of the scenarios, we engaged the participants in actually physically performing the tasks *in-situ*, rather than just imagining themselves in the respective environments and situations. For example, we had participants go into a library to look for a book (S4 below), and ride a stationary bicycle (S5). To better control conditions, we *simulated* the sensor data that Laκsa used for each scenario. The data is based on previously recorded real data of several users performing the respective scenarios. We asked participants to wear pants with front pockets to facilitate placing the mobile there and to allow comfortable cycling. To further control for experience, we provided participants with a fixed set of personal availability rules (6 rules specifying availability as Unavailable and Semi, and any other case as Available; *e.g.*, if the user is in her office and Laκsa hears talking, then she is seen as Unavailable).

We employed seven scenarios to span three situational dimensions: (i) Exploration / Verification (S1, S2, S5) where Laκsa behaved appropriately and participants are asked to explore and verify the interface; (ii) Fault Finding (S3, S4, S7) where Laκsa apparently or actually behaved inappropriately, and participants had to debug what really happened; (iii) Social Awareness (S6, S7) where participants investigated information and explanations about hypothetical buddies that they naturally had less awareness of. We measured their desire to contact their buddy in the latter scenarios to gauge their trust of Laκsa.

S1: Sitting in office talking. Training session where the participant learned Laκsa's core features and explanations.
S2: Walking outdoors. Verification/exploration task where participants investigated explanations of Place and Motion.



	S2 Walking Outdoors	S5 Cycling in Gym	S7 Talking as Music	S6 Friend Available	S3 Missed Contact	S4 Library Interruption
🟥 Gave up / Failed						P13
🟧 Misunderstood	P3	P3	P4		P1 P3 P4 P6 P7 P8	
🟨 Struggled to understand	P1 P9		P3 P6 P8 P12	P10	P5 P11 P13	P1 P4 P7 P8 P12
🟩 Effectively understands	P5 P6 P11 P13	P1 P5 P6 P7 P9 P12		P5 P7		P5 P6 P10

Exploration / Verification
Social Awareness
Fault Finding

Table 2. Participant results in scenarios showing how each participant performed for each scenario as he or she used explanations provided in Laκsa. Note that columns are not arranged in categories, not by sequence of presentation.

S3: Missed contact. The participant was asked to find out why she was considered not available to a friend, even though she was. This is a false-positive error condition where Laκsa actually behaved correctly, but given that availability is multi-faceted *(i.e.* different statuses to different viewers), the participant may not realize this.
S4: Library interruption. The participant walked to a nearby library to search for a specific book. She needed to squat to retrieve the book on the lowest shelf. As she was looking for the book, the phone rang (the experimenter invoked a ringtone in the library), indicating a call from a coworker. The participant was asked to determine why she was not seen as unavailable. This is a true-positive error condition where we simulated Laκsa making a mistake.
S5: Cycling in gym. Exploration task for participants to explore how Laκsa tracks their cycling in a gym location.
S6: Friend available. The participant was asked to check Laκsa to help decide whether to contact a friend.
S7: Friend talking inferred as music. The participant was asked to find out why, when the friend was actually in a meeting, Laκsa made the error of telling her that the friend was Available, at the office listening to music, and had nothing scheduled.

Participants

Using a local recruiting website, we recruited 13 participants (8 females) with a mean age of 26.4 years (range: 18 to 37). P2 was dropped because he did not continue beyond the training scenario. Four participants were students (one undergraduate). Only P1 was trained in a computer-related field (information systems), while the others spanned a wide range of areas (*e.g.*, rehabilitation, mathematics, materials science & engineering, human resources). We engaged each participant for 2.5 hours on average (range: 2 to 3.5). Due to the length of each scenario, each participant experienced between 3 and 6 scenarios (median=4), selected to try to balance coverage. Participants were compensated $10/hr.

DATA ANALYSIS

We transcribed the think aloud and interview data, segmented by speaker (interviewer / interviewee). The transcript was coded by question type (*e.g.*, Why, Why Not, What If), goal / intention / rationale (verified during interviews), feature requests, breakdowns / struggles (*e.g.*, too many questions to choose from), and extent of (mis)understanding. We formed sequence models of the usage of question types, and consolidated them (*e.g.*, see Figure 4). We interpreted the findings and models to identified causal factors. This was assembled into higher-level themes using an affinity diagram (selection based on theme convergence, importance, and novelty).

Next, we describe our observations of how participants used Laκsa, focusing on how they asked for various explanation types. Table 2 summarizes how much participants understood Laκsa's inference in each scenario.

PATTERNS OF INTELLIGIBILITY USE

We found different usage of explanations for the three different situational dimensions presented in the scenarios: exploration/verification, social awareness, and fault finding.

Exploration / Verification

We observed that participants explored all explanation types as they tried to *learn* how Laκsa functioned, and more about the scenarios. Naturally, participants used the Description explanations to *remind* them what the terms and concepts meant, and used the Inputs explanation to *examine* deeper states that affect inferences (*e.g.*, for S3, P11 used the Inputs explanation of Availability to see a "summary of her statuses"). How To explanations were used in two ways: P11 appreciated learning new concepts about inferring Sound (through periods of silence, pitch ranges, *etc.*); for S3, P6 checked to see when she would be Unavailable. Finally, some participants asked What If to *preemptively* test troublesome or critical situations to see how Laκsa would respond under those circumstances; *e.g.*, for S3 during a lull period, P3 and P5 confirmed that they remained available to family members in an emergency.

Social Awareness

For S6 and S7, participants had less *first-hand* knowledge of the actual situation about their buddy than about themselves. We observed that they mainly focused on the Input explanation of Availability (or equivalently, What explanations of the lower-level contexts). For example, P6 first checked the Input values of Availability to determine the state of her friend (in S7). Participants would then form *stories* about what they believed their buddy could be doing at the time; *e.g.*, having seen the Motion inferred as "Other" (placed flat) for S6, P9 presumed his friend was "probably sleeping or taking a nap or eating or something that would probably involve not wanting a phone call." When he subsequently looked at the Sound inference, he said "sound is 78%, oh he is listening to music, so I guess I could intrude if I want to, but then I get he might be with someone else too." At this point, most participants did not continue by asking other questions. However, for S6, P7 asked When would her friend be Semi-Available (How To) to learn the rules her friend had set.

We found that once participants perceived that the availability status was inferred inaccurately or erroneously, they explored other questions and investigated more deeply. This is similar to how participants investigated anomalies with explanations about themselves (discussed next).

Fault Finding

Participants used explanations most when perceiving that Laκsa behaved *unexpectedly*. Figure 4 summarizes the sequences of how participants asked for various explanations (labels refer to observations described in the following text). The choices of questions were slightly different when investigating the top-tier Availability than the lower-tier contexts (Place, Motion, & Sound).

Top-tier, Rule-based Availability

When participants realized that the Availability was wrong, some *instinctively* first selected the Why question (Figure 4: 1). P6 first asked why about her Availability in S3 and S4, and why her Place was inferred as Office in S4. When explaining her rationale to first select why (S4), P12 said "because I want to know *why…why* I'm available." This suggests a *linguistic cue* for asking Why first.

Alternatively, sometimes participants first *inspected* the state of the application by asking for Input values (2). For S3, by examining the Input values, P11 discovered that "the Talking and possibly the Work [category] in the schedule [were] the two that led my status to be unavailable." For S4, P6 and P12 quickly discovered that the Place inference (as Office) was erroneous (should be Library instead).

If participants had an *expectation* of what the availability should be, they would ask about the expected outcome using Why Not or How To questions (3a, 3b). These represent different strategies to address this goal-oriented query. Interestingly, some participants asked How To instead of Why Not, even though the latter was more *concise*; *e.g.*, for S4, P7 asked When would status be Semi-Available (How To) to *manually* identify erroneous conditions out of three. Had she asked "Why isn't status Semi-Available", she would have seen the single condition that was specifically identified to be relevant to the scenario. Unfortunately, some participants *misunderstood* the How To explanation, *e.g.*, for S7, when P4 asked "When would Sound be Unavailable," he interpreted the requirement that his friend has to be Talking to represent that his friend was currently sensed as talking; for S4, P7 examined the rules for when she would be Semi-Available (as she had expected status to be), found the condition Place = Library, and misunderstood that to mean that Laκsa had correctly inferred her location. Another reason for the lack of use of Why Not could be that the explanations with *contrapositives* put off users from using Why Not more: P6 complained about the "excessive use of negatives."

Some participants *simulated* conditions they expected to be true with the What If explanation (4), to see if Laκsa would infer an expected Availability. For S4, P8 asked What If, setting Place to Library (which she believed to be ground truth), and Ringer to Silent (which she believed she should have set). This resulted in the status of Semi-Available, which was correct unlike the actual inference of Available, and indicated that while the rule was executed correctly, possibly something was sensed wrongly. Unfortunately, participants were prone to *carelessness*: while setting up the expected state for S2, P6 changed three contexts (Place = Café, Motion = Sitting, Contacter = Friends), but failed to notice her schedule was set to Work (a pivotal factor to determine her status to friends). She expected her status to appear as Available, but it appeared as Unavailable instead; for S4, P13 forgot to set Place to Library (left as inferred value Office) when trying to verify the inferred availability.

Using the aforementioned strategies, some participants may decide to add a new rule or modify one to fix the anomaly, and this may be a satisfactory solution. However, most of the problems in the scenarios are due to faults at the lower-tier context inference. To investigate further, they would select the suspect context by clicking on the respective tab.

Lower-tier, Sensed / Inferred Contexts

Once again, participants *instinctively* asked Why (5): *e.g.*, for S2, P9 asked Why to see "that map thing," the Map visualization showing which place his location overlapped with; for S2, P6 asked Why Motion was inferred as Walking, only paying attention to which features were listed, and not paying attention to the boundary conditions; for S3, P6 first asked Why Sound was inferred as Talking, but found that unhelpful.

Participants also paid attention to the inference Certainty (6). For S6, after noting a Sound certainty of 73%, P10 mentioned that as long as it was above 50%, that was "good enough." For S7, P12 accepted the inference for Sound as Music (93% certainty), because it was above 90%. Subsequently, he was confused when this inference turned out to be wrong. When in doubt of the current inference, some participants also made it a point to find out which other Output values were *plausible* through their Certainties (7); *e.g.*, for S6, P6 checked the certainty of inferring Sound as Talking (8%), grew wary that the actual inference (Music at 73%) may be wrong, and became hesitant to contact his buddy. For S4, P12 wanted to see whether Laksa recognized squatting, and seeing Certainties of 67% for Standing and 33% for Cycling, he accepted that "cycling is kind of like squatting" and "partially standing."

When asking about an *expected* outcome or exploring an *alternative* outcome with a noticeably high certainty, participants similarly demonstrated two dominant exploration strategies: asking Why Not (8a), or How To together with Inputs (8b). Asking Why Not provides a concise explanation that directly compares the actual inference with the desired outcome. For S4, immediately after noticing a wrong Place inference, P5, P6, P12 asked Why Not to see why their location bubble did not overlap with the bubble for Library. For S5, P5 used the Why Not Physical Diagram to explore how Running was not inferred (Cycling was). For S7, P6 became uncertain after noticing, from the Weights of Evidence visualization, that Pitch Fluctuation strongly voted for inferring that her friend was Talking, while every other factor voted for Listening to Music (see Figure 3d). However, we found that many participants disliked the explanations as being *too technical*, particularly, the Physical Diagrams for Motion. In fact, for S5, focused on Cycling, P7 avoided the Motion diagrams of the Input, Why, and Why Not explanations. For S6, P9 found the Weights of Evidence visualization explaining "Why isn't (Why Not) Sound Ambient Noise" confusing (difficult to remember what icons meant), and preferred to just look at the Output Certainty of Ambient Noise. Other participants alternatively used the How To explanation in conjunction with Inputs, by *manually* comparing the two explanations; *e.g.*, for S6, P9 repeatedly toggled between the How To and Inputs metaphorical diagrams for Sound. He studied Pitch Pureness to see if the current Input value (=29) was "just about right" compared to the average value for Music (=34), and accepted the Sound inference of Music. After several exposures to both techniques, P12 realized (in S5) that the Why Not explanation for Motion provided similar information as How To + Inputs, and required less effort to inspect.

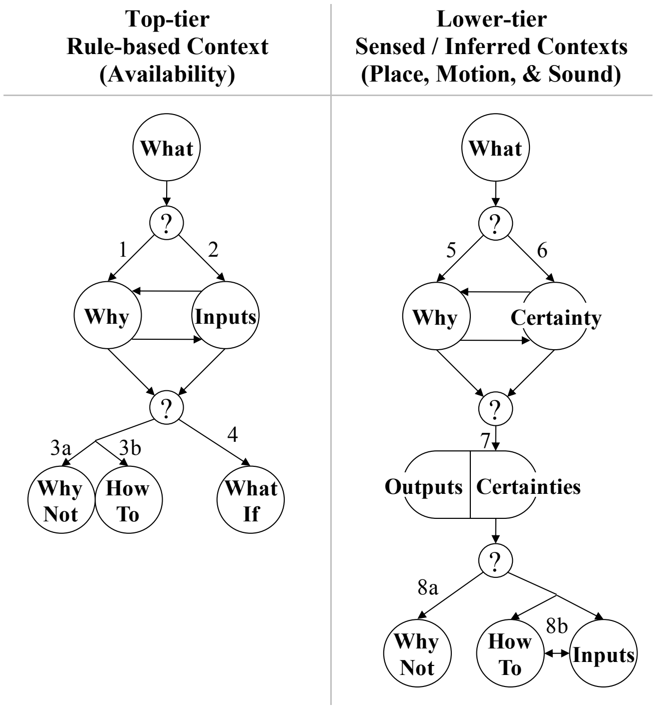


Figure 4. Consolidated sequence models of explanation use for fault finding. Participants chose different question type explanations and in different sequences. '?' indicates participant desire to ask questions; arrows indicate transitions of using each explanation; only prominent behavior is noted.

THEMES OF INTELLIGIBILITY USE

We created an affinity diagram of our coded findings to map out core issues and patterns of use. Here, we present the top three high-level themes of how, and why participants used or failed to use the intelligibility features.

Information Overload and Explanation Detail

While we intend explanations to express a comprehensive view of what Laκsa knows and how it infers, we run into the problem of *information overload*. Comprehensive explanations are too long and complicated for end-users. Even though we took several steps to reduce the explanation complexity, participants still complained about the remaining complexity. P1 pointed out (as expected) that there were too many questions to choose from, and she did not necessarily know which was best for her goals. P3, P7, P8, and P9 also complained about the large number of reasons provided for Availability explanations (up to 9), and the large number of Input features described for Motion and Sound. In fact, P3 suggested showing up to 3-4 reasons, most participants only paid attention to 1-3 features of Motion (especially just vigorousness and movement), and Sound (periods of silence, pitch range, pitch pureness). When trying to determine her friend's availability for S6, P10 grew tired of asking for explanations. She felt "like it was information overload," and that she "started getting less information about what he was doing." She started doubting whether her friend was actually listening to music or sleeping instead. Clearly, our lay users did not want a lot of explanation detail.

Furthermore, participants preferred Certainty explanations because of its single value (*e.g.*, P9). Similarly, P12 eventually showed a preference for the more concise Why Not instead of How To explanation for explaining Sound. While participants found the Motion and Sound Input feature details interesting, they also found them too technical for such a lay-user application (*e.g.*, P3, P7).

Prior Knowledge and Relatability

It is well-known that *prior knowledge* plays a role in learning and understanding, and we observed how that influenced how participants used and interpreted explanations. For example, since Laκsa uses access points to sense location, the geographical distribution of these points affects the inferred location, and the distance error. However, P4 did not know this, and had no idea in S4 that Place was wrongly inferred as not being the Library.

This problem was more widely manifest in a usability issue: some participants wanted to see explanations framed in terms *relatable* to their activity; *e.g.*, for S5, P1 wanted Motion vigorousness to be stated as "within the walking or running range" or in terms of exercise "high or low intensity". She found forces and orientation unhelpful to understand her exercise. While this can clearly help users in making sense of input values, this is less feasible if an input has multiple ranges of values for certain outcomes (*e.g.*, multiple ranges of orientation angles for sitting due to different resting angles of the phone in a pocket).

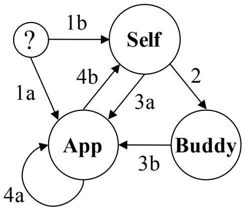


Figure 5: Blame shifting during S4 about mistakenly receiving a phone call in the library.

The participants' use of Inputs and How To explanations for a comparative method to derive a Why Not explanation also suggests this need to frame sensor features in terms of well-understood activities. For S7, to convince themselves that a sound heard was really talking, participants looked at the feature values in Inputs (representing the current state), and values typical of Talking (found by asking How To).

Different Strategies in Problem Solving

We observed that some participants employed *suboptimal* problem solving strategies to try to determine how Laκsa made inferences. We observed a lack of strategy and logical fallacies such as causal oversimplification. Even when provided with various explanation tools, some participants did not know how to effectively use them. For S2, P11 *sequentially* explored questions in the drop-down list of questions, while others (*e.g.*, P6, P12) chose the Why explanation *instinctively*. Many participants also asked How To to get a Why Not explanation. This required them to do *manual* work to identify which reason was relevant, when Why Not would have automatically selected it.

Participants also exhibited common logical fallacies. Many participants exhibited *causal oversimplification* [3], because as they looked at reasons (*e.g.*, from Why, Why Not), they mistakenly fixated on a single factor and ignored others. P6 and P11 felt that Schedule was the "explicit way" of saying whether they were available (S2). P4 thought that since his friend was at the office (S7), then he must be Unavailable, even though he also would have needed to be talking. Having formed this wrong belief that his friend must be busy, P4 persisted in looking for clues to verify his hypothesis, rather than revise it. He was thus unable to identify the problem without help from the experimenter.

Blame Shifting

We observed participants *shifting blame* attribution in several scenarios, particularly, S4 where the participant received a loud call while at the library (see Figure 5). Participants changed who or what they attributed blame to after carefully considering what they knew, and viewing Laκsa's explanations. Immediately after receiving the interruption, P1, P7, P8, P12 reacted instinctively and were annoyed with Laκsa (1a), while P6, P13 were self-judgmental and felt they forgot to silence the ringer appropriately (1b). On reflection, P7, and P12 realized their coworker should have seen their Place and known not to call them. They would then blame their coworker for violating social norms (2). After looking at the availability status displayed, participants realized Laκsa was indeed wrong, and *all* participants shifted blame to Laκsa (3a, 3b).

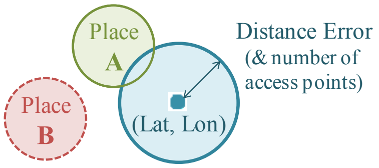


Figure 6. Simple "bubble" components simultaneously for explaining seven questions about Place. *What*: Place inferred as at place A. *Inputs*: (Latitude, Longitude) coordinates of sensed current user location. *Output possibilities*: place A, B, *etc. Why at A*: because sensed location bubble overlaps with bubble of A. *Why Not at B*: because sensed location bubble does not overlap with B. *How To be inferred at B*: need location and B bubbles to overlap. *Certainty of inference*: distance error and number of access points.

However, after viewing explanations and finding out that the status error was due to a poor location sensing (and both Library and Office in tight proximity), participants had different reactions. P8 and P12 continued to blame Laκsa for its imprecise sensing, and even became harsher in their judgment, because they (incorrectly) expected indoor location sensing to be as precise as contemporary GPS devices (4a). On the other hand, P6, P7, and P13 forgave it because they understood how challenging it was to sense location in the given circumstance, and understood that they would have to change a setting to improve sensitivity (4b). This supports findings about reduced blame attribution when autonomous robots explain their actions [6]. While one might assume explanations improve perception and trust of Laκsa, this observation reveals explanations to be a double-edged sword: revealing the difficulty of some situations, or exposing a lack of competence.

DESIGN RECOMMENDATIONS FOR INTELLIGIBILITY

Drawing from our design exploration and user study, we present some recommendations on how to improve the usability of intelligibility that can be applied more generally to context-aware applications.

Reducing and Aggregating Explanations

We had originally employed this strategy before the user study when designing Laκsa, and found this to be even more crucial based on our study results. In fact, participants demanded an even lower level of detail, wanting to see more details only as needed. Hence we continue to recommend this requirement. One compromise would be to allow incremental access to more detail *on demand* and offer significantly reduced explanations initially. Alternatively, it could be even better to present them in a form that is concise but does not compromise by omitting any reasons. Finding How To explanations cumbersome due to the large number of tabs to see multiple reasons, P1 suggested just presenting the rules in a table instead. This would provide a *bird's eye view* of the rules and yet be much easier to access. Furthermore, because some participants found some features for Motion and Sound to be overly technical, it may be sufficient to filter them out of explanations rather than make them physically meaningful, or provide metaphors to explain them. However, it is unclear whether users want to see them when encountering more serious and esoteric debugging problems.

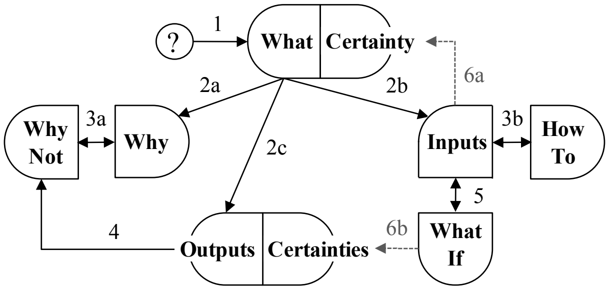


Figure 7. Streamlined sequence diagram for explanation use.

Retooling Explanations with Simpler Components

Several participants (*e.g.*, P6, P9, P13) referred to explanations of Place as "the bubble thing" or "map thing" instead of noting which question they wanted to ask. They used the simple bubble components of Place (Figure 3a) to investigate various questions (Figure 6). Therefore, it may be better to design simple explanation components that can be used to answer multiple questions than to individually answer those questions through different automatically generated representations: *i.e.*, use explanation components with a smaller *vocabulary* set expressive enough to convey most of the explanation types we have employed. Unfortunately, it is difficult to design reusable, simple explanation components for contexts like motion and sound, because they depend on a wider and more diverse range of features that may not be represented equivalently.

Streamlining Questioning

Aware of her lack of understanding to effectively use the explanation questions, P7 suggested a *flow chart* to help guide users to ask optimal questions. We propose the *streamlined* flow of questions as shown in Figure 7, where only 1-3 question types are accessible at any given time. This helps reduce information overload when choosing explanations. Users first start with seeing What the application has inferred, along with its Certainty (1). If they want to ask questions, they can seek the mechanistic rationale by asking Why (2a), explore the system Inputs state (2b), or explore the Certainties of alternative Output values (2c). If users want to know why an expected or alternative output was not inferred, they may ask Why Not (3a), compare Inputs with How To (3b). These support the two observed why-not strategies. Having observed how participants wanted to ask questions from the Laκsa UI, we recommend convenient shortcuts: users can ask Why Not on seeing alternative Output values (4), and simulate different Input values to ask What If (5). Finally, users can also explore the values (6a) and Outputs (6b) of lower-tier contexts through Inputs and What If, respectively.

Non-Mechanistic Explanations

We found that users need more types of explanations to properly ground them in the application domain, and to educate them about good problem solving and debugging strategies to fully understand the program functionality.

Given the deep knowledge that complex context-aware applications rely on to make decisions, and the evidence that some participants lacked sufficient prior knowledge to relate technical behavior to real-world and domain phenomena, we can see that simple textual descriptions are not sufficient to scaffold the automatically generated explanations. This suggests context-aware applications need to have access to information about complex real-world concepts that are not necessarily core to the application.

Moreover, we found that since some users did not effectively leverage the explanation facilities provided, intelligible applications may need to teach them problem solving strategies. One solution may be to provide examples of end-user debugging with the explanation tools.

LIMITATIONS AND FURTHER WORK

We used an iterative design process where design decisions were based on careful consideration, consultation with HCI experts, and user feedback. While we believe our designs are reasonably interpretable, an alternative approach is to make comparisons between competing designs.

We have limited our study to a manageable set of scenarios, chosen to aid exploration of explanation use, rather than to comprehensively cover situations. Hence, our results are suggestive rather than definitive, and we seek to validate them with further design iterations and user studies.

Our design recommendations are based on studying how and why users seek explanations given several goal situations. While we expect following them would improve the usability and thus usage of explanations, they may not be the best to *promote understanding*. Future work will explore how to design and provide explanations that are not only easier to interpret, but also effective in improving the understanding of how context-aware applications work.

CONCLUSION

We have described our first steps to building a real-world, intelligible mobile context-aware application. We followed several design principles to improve the usability of explanations, and conducted a user study to discover how users make use of explanations, and issues they experienced. Particularly, we investigated the use of intelligibility for the mobile contexts of Availability, Place, Motion, and Sound activity. Our findings emphasize the importance of making explanations usable and quickly consumable (by reducing information overload), relating the application behavior to the real world activity (to raise the relevance of the information), and supporting effective problem solving and debugging strategies (so that users can quickly understand the application issues before giving up). We suggest a need for streamlining explanations while maintaining access to the rich explanation capabilities, and for integrating domain knowledge in explanations. With a better understanding of how users use the question type explanations, we can better design explanations to help understand sophisticated context-aware applications.

ACKNOWLEDGMENTS

This work was funded by the National Science Foundation under grant 0746428, and the Agency for Science Technology And Research, Singapore. We thank for early development work: Zhiquan Yeo and Kanupriya Tavri. We thank for helpful advice: Scott Davidoff, Bryan Pendleton, Leonghwee Teo, Karen Tang, Eiji Hayashi, John Santerre, Chris Harrison, Matt Lee, Chloe Fan, and Aubrey Schick.

REFERENCES

1. Bellotti, V. & Edwards, W.K. (2001). Intelligibility and Accountability: Human Considerations in Context-Aware Systems, *Human-Computer Interaction*, 16(2-4): 193-212.
2. 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.
3. Damer, T.E. (2009). Attacking Faulty Reasoning: A Practical Guide to Fallacy-Free Arguments (6th. ed.). Belmont, CA. Wadsworth Cengage Learning.
4. 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.
5. Dey, A.K. & Newberger, A. (2009). Support for context-aware intelligibility and control. *CHI 09*, 859-868.
6. Kim, T. & Hinds, P.J. (2006). Who should I blame? Effects of autonomy and transparency on attributions in HRI. *ROMAN 06*, 80-85.
7. Kulesza, T. *et al*. (2009). Fixing the Program My Computer Learned: Barriers for End-users, Challenges for the Machine. *IUI 09*, 187-196.
8. Lester, J. *et al*. (2009). A Practical Approach to Recognizing Physical Activities. *Pervasive '06*, 1–16.
9. Lim, B.Y. *et al*. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. *CHI 09*, 2119-2128.
10. Lim, B.Y., & Dey, A.K. (2009). Assessing Demand for Intelligibility in Context-Aware Applications. *Ubicomp 09*, 195-204.
11. Lim, B.Y., & Dey, A.K. (2010). Toolkit to Support Intelligibility in Context-Aware Applications. *Ubicomp 10*, 13-22.
12. Lu, H. *et al*. (2009). SoundSense: scalable sound sensing for people-centric applications on mobile phones. *MobiSys '09*, 165-178.
13. Miluzzo, E. *et al*. (2008). Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. *SenSys 08*, 337-350.
14. 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.
15. Oulasvirta, A. (2005). Grounding the Innovation of Future Technologies. *Human Technology* 1 (1):58-75.
16. Tullio, J. *et al.* (2007). How it works: A field study of non-technical users interacting with an intelligent system. *CHI 07*, 31-40.
17. Vermeulen, J. *et al.* (2010). PervasiveCrystal: Asking and Answering Why and Why Not Questions about Pervasive Computing Applications. *IE 10*, 271-276.
18. Ware, C. (2000). *Information Visualization: Perception for design*. San Francisco, CA: Morgan Kaufmann.
19. Weiser, M. & Brown, J.S. (1997). The coming age of calm technology. *Beyond Calculation: the Next Fifty Years*, 75-85.