SenSay: A Context-Aware Mobile Phone

Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Neema Moraveji, Kathryn Reiger, and Jeremy Shaffer
Human Computer Interaction Institute and Institute for Complex Engineered Systems
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{dps, asim}@cs.cmu.edu, {junichif, nmoravej, kr, jshaffer}@andrew.cmu.edu

Abstract

SenSay is a context-aware mobile phone that adapts to dynamically changing environmental and physiological states. In addition to manipulating ringer volume, vibration, and phone alerts, SenSay can provide remote callers with the ability to communicate the urgency of their calls, make call suggestions to users when they are idle, and provide the caller with feedback on the current status of the Sensay user. A number of sensors including accelerometers, light, and microphones are mounted at various points on the body to provide data about the user's context. A decision module uses a set of rules to analyze the sensor data and manage a state machine composed of uninterruptible, idle, active and normal states. Results from our threshold analyses show a clear delineation can be made among several user states by examining sensor data trends. SenSay augments its contextual knowledge by tapping into applications such as electronic calendars, address books, and task lists. The phone alleviates cognitive load on users by various methods including detecting when the user is uninterruptible and automatically turning the ringer off.

1. Introduction

Current commercial mobile phones impose additional cognitive load on their users by requiring them to be conscious of their phone's states. Examples include remembering to turn the ringer on and off, handling missed calls, determining call priority, and worrying about inaudible ringer volume in a loud environment. This paper attempts to alleviate users of these inconveniences, creating a phone that can adapt to user's context changes.

SenSay (*sen*sing & *say*ing) is a context-aware mobile phone that modifies its behavior based on its user's state and surroundings. It adapts to dynamically changing environmental and physiological states and also provides the remote caller information on the current context of the phone user. To provide context information SenSay uses light, motion, and microphone sensors. The sensors are placed on various parts of the human body with a central hub, called the sensor box, mounted on the waist (see Figure 1).

Figure 1. SenSay: sensor box mounted on the hip (left), the mobile phone (center), and voice and ambient microphones mounted on the user (right).

SenSay introduces the following four states: Uninterruptible, Idle, Active, and the default state, Normal. A number of phone actions are associated with each state. For example, in the Uninterruptible state, the ringer is turned off.

Some related work is reported in the following papers. In a much more limited context the idea of smart appliances and phones was explored in [1], [2], [4], and [5]. In [3] concepts of context-aware computing and wearable devices have been described.

2. SenSay Architecture

2.1 General Overview

A closed architecture was adopted with five functional modules: the sensor box, sensor module, decision module, action module, and phone module. The following components are shown in Figure 3, from left to right: the sensor box collects physical sensor data, the software-based sensor module queries that data, the decision module determines the phone's state, the action module sets that state, and the phone module provides access to the mobile phone operating system and user interface.

In the current prototype, the decision, sensor, and action modules run on a notebook computer (henceforth called the

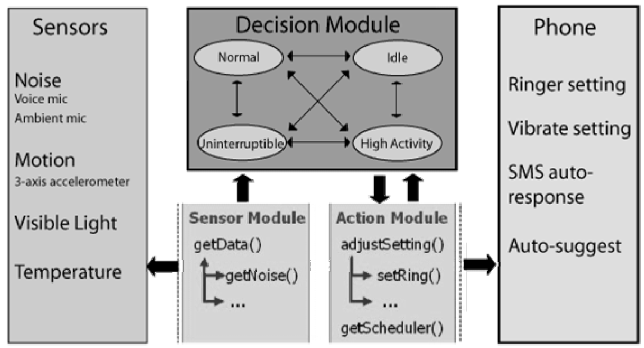

Figure 2. SenSay architecture depicting the 5 modules.

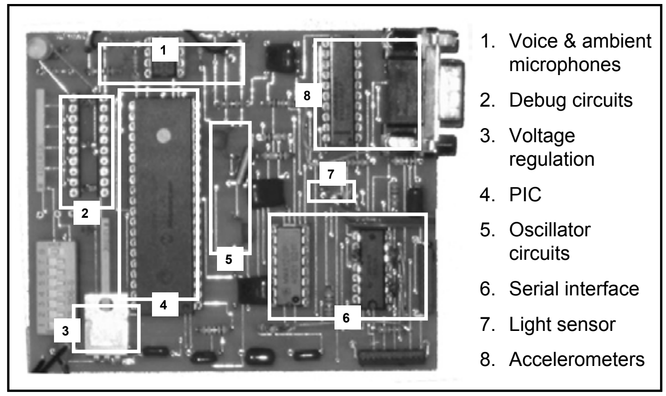

Figure 3. Layout of the sensors on the PCB.

platform) running MS Windows 2000, which is connected to both the sensor box and the mobile phone via RS232 serial connections. In the next revision of SenSay, the mobile phone will include the sensor, decision, and action modules and communicate directly with the sensor box.

2.2 Sensor Box

The sensor box includes a printed circuit board (PCB), as shown in Figure 3, housed in a plastic casing. The board circuitry consists of two subsystems, the sensors and the microcontroller. A PIC16F877 microcontroller is the heart of the sensor box. It provides eight 10-bit analog-digital conversion (ADC) channels as the interface to the sensors, as well as a port for serial communication to interface with the sensor module. The microcontroller is used to process the queries from the sensor module and return the requested sensor data as a 10-bit word.

The specific sensors integrated in the sensor box are shown in Table 1. The two microphones are connected to the board through long (~1m) wires to allow mounting on various parts of the body. The light sensor is also connected through a long wire allowing it to be mounted on the phone. Three accelerometers are used to capture three (x, y, z) degrees of motion and are mounted directly on the PCB. Various support circuits are also included to condition the output signal of the sensors to insure the outputs remain in the desired range.

2.3 Sensor Module

The sensor module is responsible for querying the sensor box periodically (currently once per second) and returning that data to the decision module. A simple communication protocol has been defined, supporting basic query/response commands.

Communication between the microcontroller and the sensor module is established through the standard RS232 serial connection. A unique ID number identifies each

Hardware Component	Relevant Specifications
MAX233-RS232 Connection Maxim MAX233	12 volts transmission rate, internal capacitors, 5v supply voltage, 20-pin DIP housing, 120kbps communication bandwidth
Voice Microphone Emkay Innovative Products SP0101NC2-2	omni-directional, 100Hz - 10KHz frequency range, 3-5Vdc supply, -42dB re 1V/Pa sensitivity, highly shock resistant
Ambient Noise Microphone Emkay Innovative Products MD6020ASC-0	omni-directional, 100 Hz – 10 KHz frequency range, 2-10Vdc supply, -42dB re 1V/Pa sensitivity
Accelerometers Motorola MMA1201P Motorola MMA2200W	captures 3 axes of motion: outputs range from +/- 38 g, output changes by 50mV/g, basic op-amp circuit changing the reference point to 2.5V to scale output voltage
Temperature National LM35	linear output from base temperature +10mV/°C, with accuracy of +/0.5°C, range of +2°C to +50°C
Visible Light Panasonic PNA4603H Photo IC	high sensitivity to visible light, output varies linearly from 0 – 4 volts

Table 1. Specifications for sensors mounted in sensor box.

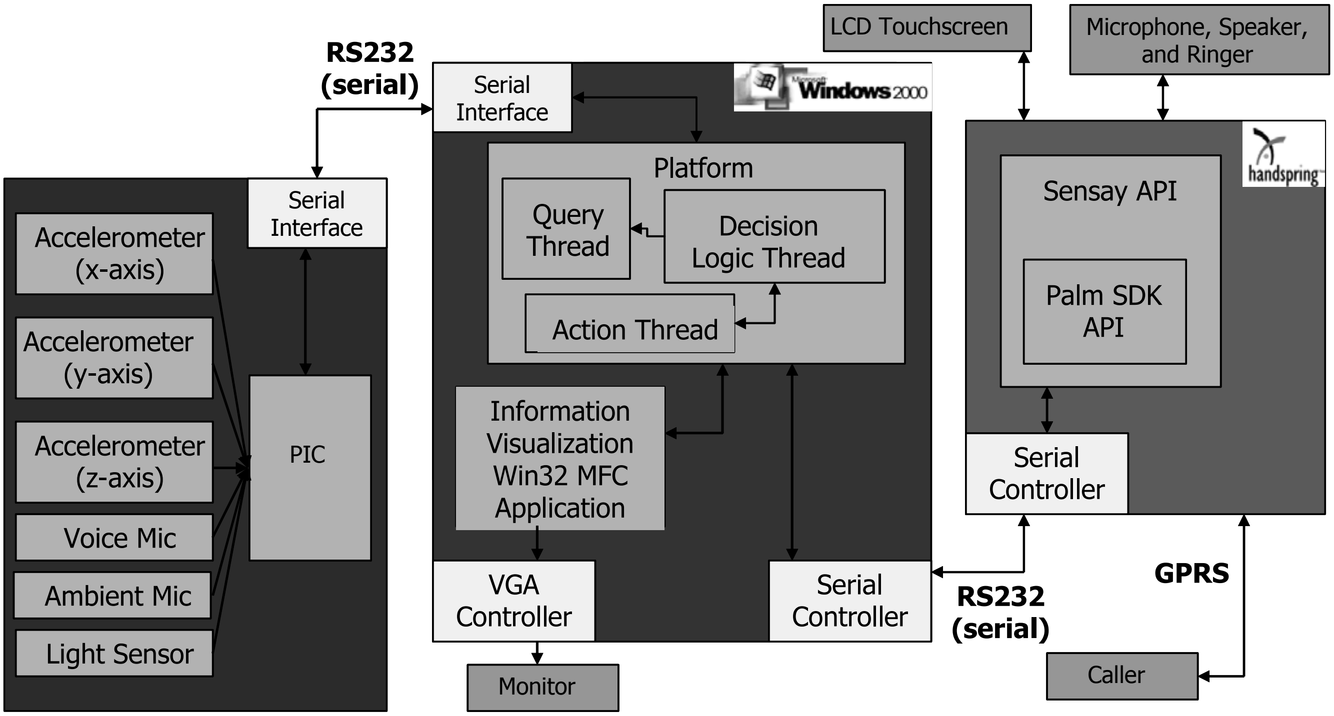

Figure 4. Architecture of the SenSay system from sensor box to platform to mobile phone.

sensor in the sensor box. At any time, the platform will query the sensor box for a particular sensor data by sending only the requested sensor's ID in a message packet. This protocol keeps the communication overhead low and also reduces the message parsing overhead on the microcontroller. The sensor box polls its sensors periodically as quickly as it can, storing the most current values in an array. When a query is processed, the microprocessor will respond with the most recent reading of the requested sensor without actually polling the sensor, but rather doing a table lookup. This allows the microcontroller to respond immediately after receiving the query.

The return packet defined by the protocol is also kept simple. The sensor responds with the sensor ID followed by the 10-bit word of sensor data, which is sent as two bytes padded with zeroes. Accelerometer data consists of three words of data, requiring a total of 6-bytes of data to be sent. For future sensors with n bytes of data, the number of bytes allocated for the sensor data will be adjusted accordingly. The sensor data size must be predefined in the protocol since the size of packet is not communicated. Currently all sensor data is requested at the same rate, however the protocol allows for requests to be sent at any time. Therefore it is possible to request data from a sensor like the microphone, where the volume changes often, at a fast rate (< 1 sec) and poll sensors that change less often at a slower rate (1 min).

2.4 Decision Module

The decision module provides the bulk of the logic used in state management. It queries sensor data and the electronic calendar of the user. Based on these inputs, this module determines what state the user is in and issues phone actions.

As shown in Figure 4, the decision module acts as a conduit between the settings on the phone and the data gathered by the sensor box. From left to right in Figure 4: the sensor box acts as the "eyes, ears, and skin" of the system, the Decision module acts as the "brain", and the phone itself acts as the voice, expressing the state of the dynamic system.

Figure 5 shows the decision module's state diagram. Ten minutes of sensor value history is stored to help determine the user's state. Therefore, within the decision module, there is one circular buffer capable of storing 10 minutes of data.

The sensor value buffer holds primarily sensor values but also other information such as previous phone state transitions and events. This will enable the SenSay team to conduct future research into the best way to utilize state transition patterns to determine current state. Each record also stores a value indicating whether or not the state command was ignored by the phone. This would occur for, example, if the user is speaking on the phone at the same time as when the decision module is recommending the Uninterruptible state – in this case, the phone ignores the decision module so the user is interrupted.

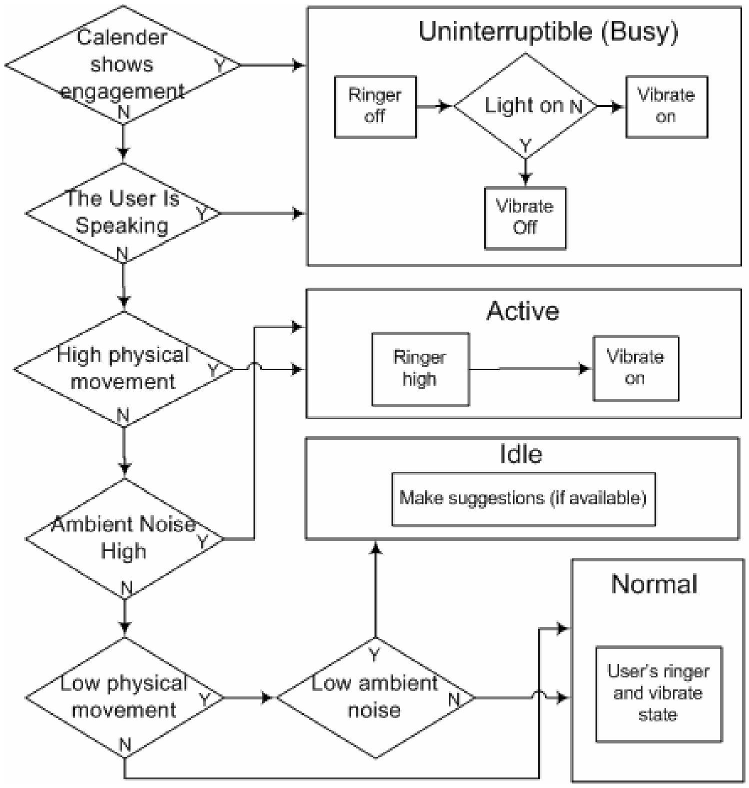

Figure 5. State diagram showing what happens when an incoming call is made when the phone is in Uninterruptible mode.

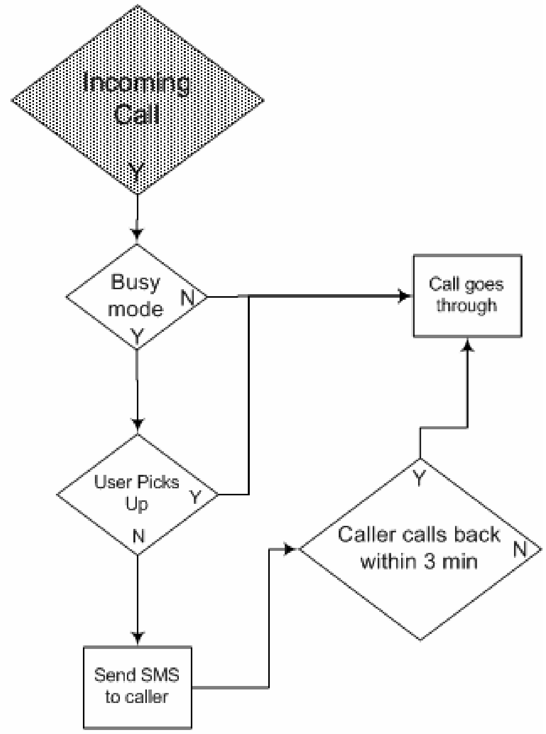

Figure 6. State diagram showing how the phone determines which mode to place the phone in.

2.5 Action Module

The action module is responsible for issuing changes in setting and operation on the mobile phone and is controlled by the decision module. Like the sensor module, it has a generic interface which offers the decision module access to phone settings and operations. This is done through a SenSay Application Programming Interface (API), a superset of the Palm Operating System (OS) API developed to simplify accessing phone operation. This API is installed directly on the mobile phone and provides access to the underlying Palm OS API as well as user interaction. The primary purpose of the action module is to support some basic operations on the phone:

1. Ringer control: off/low/medium/high
2. Vibrate control: on/off
3. Send an SMS to a caller
4. Make call suggestions
5. Provide access to the electronic calendar

Figure 6 shows what happens when an incoming call is made to a busy user. A Short Message Service (SMS) text message is sent back to the caller informing them of the user's busy status. The SMS also informs the caller that if the call is extremely urgent, they can interrupt the user by calling back within three minutes (a configurable value). This is an example of how SenSay can empower callers to communicate the importance of their call to a SenSay user.

The Action Module triggers user interface changes on the mobile phone's LCD touch-screen display. These user interfaces communicate the phone's status to the user and permit manual override of the phone's decisions while allowing the input of preferences.

Figure 7 shows the SenSay user interface on the mobile phone for overriding the phone's state to the user's preference. Figure 8 shows a screenshot illustrating a call suggestion made while the user is in Idle state. In this case, the user's mother has called three times during the day but the user was unable to take the call. The phone was waiting until the user was in the Idle state and then suggested that the user call his mother.

3. SenSay Logic

3.1 Decision Logic

The decision module inspects all the gathered data and produces a small number of outputs, the most important of which is the current state that the phone should enter. This logic can inspect at most the past 10 minutes of sensor data.

The reason sensor history is stored is to ensure smooth transitions from one state to another, as opposed to placing the phone into a different state every few seconds. If the phone state is dictated by sensor data interpreted at each sampling interval, state transitions would be susceptible to small changes in environment data. For example, if the user sneezes, SenSay would perceive that the "user is speaking" and then when the sneeze is over (one second later), the phone would revert to the Normal state. This problem is solved since duration is taken into consideration when interpreting sensor data.

3.1.1. Uninterruptible state. If the user is engaged in a meeting or conversation, it is essential that the phone accurately assesses and reacts appropriately. The first priority of detecting the Uninterruptible state is checking the electronic calendar to determine whether or not a meeting has been scheduled. The second priority is determining if the user is speaking aloud (e.g. lecturing before students) or engaged in a conversation. The reason these two rules are prioritized is to help the system detect difficult social situations. For example, if the user is attending a lecture and is not speaking, they should still be considered uninterruptible. It is assumed that users consistently schedule appointments in their electronic calendar.

When in the Uninterruptible state, the phone also allows callers to relate the importance of their call and interrupt the user with a high-priority phone call. This is done be means of SMS. When the user is in this state and receives a phone call, the phone automatically sends an SMS message back to the caller instructing them to call back within three (3) minutes if the call is urgent. If the caller calls back within three minutes, the phone will override the current Uninterruptible state of the phone and ring at a high volume, communicating the caller's urgent context to the user.

A low threshold is required to place the user in the Uninterruptible state when the user is engaged in conversation. The reason for this is that it is better to be conservative when it comes to how much speaking is required before formally declaring a "conversation". As much as possible, the phone won't interrupt the user, thereby eliminating potential undesired inconveniences. For this reason, only the past 5 seconds of user voice and ambient noise data (the two sensors required to define a conversation) are inspected when determining if the user is uninterruptible. Conversely, a large amount of sensor history is considered before transitioning from the Uninterruptible to Normal state. The reason for this is that the preference is to err on the side of not ringing. This creates a balance where it is "easy" to enter into the Uninterruptible state (i.e., only the past 5 seconds of voice data is inspected) while being "difficult" to exit into the Normal state (i.e. 5 *minutes* are inspected to confirm that the user is no longer in a conversation).

The Uninterruptible state contains two sub-states: Light-On and Light-Off. The justification for these implied states is that when the user has the phone in their pocket (Light-Off), it is prudent to turn vibrate mode on. However, when the table is out of the pocket (on a table, desk, etc.), this is not necessarily the case because vibrations can interrupt users by creating sounds on hard surfaces such as tables. When the light sensor detects that the phone is out of the

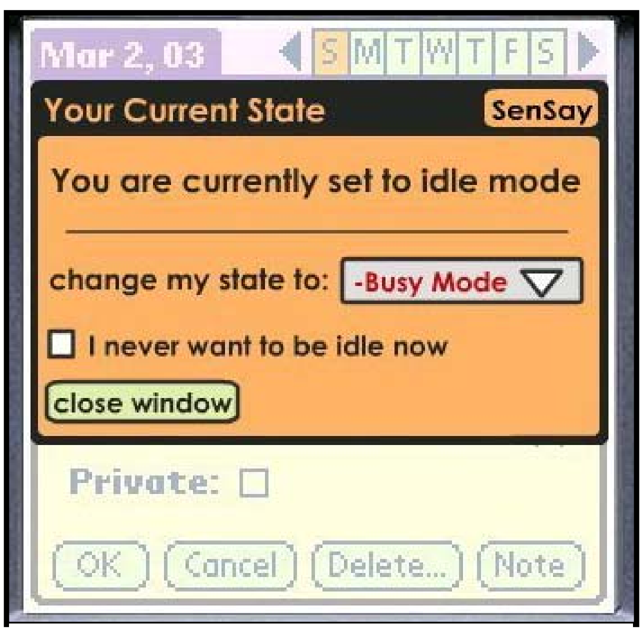

Figure 7. The SenSay user interface for manually setting the phone's state from Idle to Uninterruptible (busy).

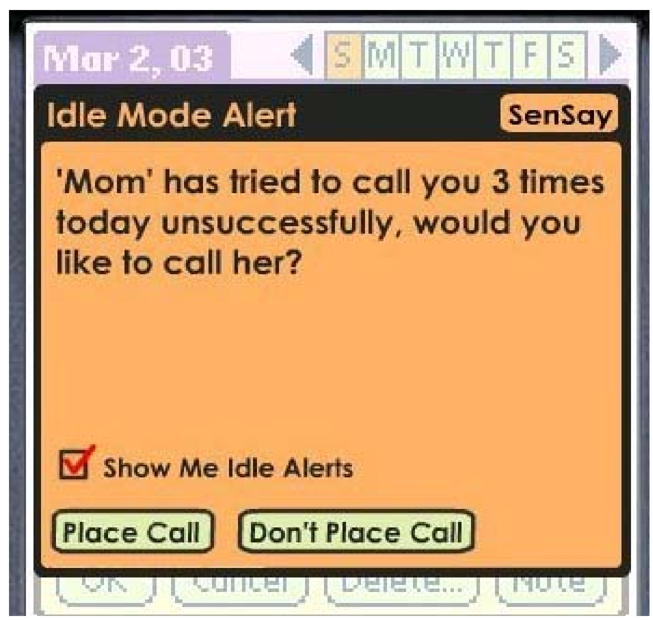

Figure 8. The SenSay user interface for displaying a call suggestion to the user during Idle mode.

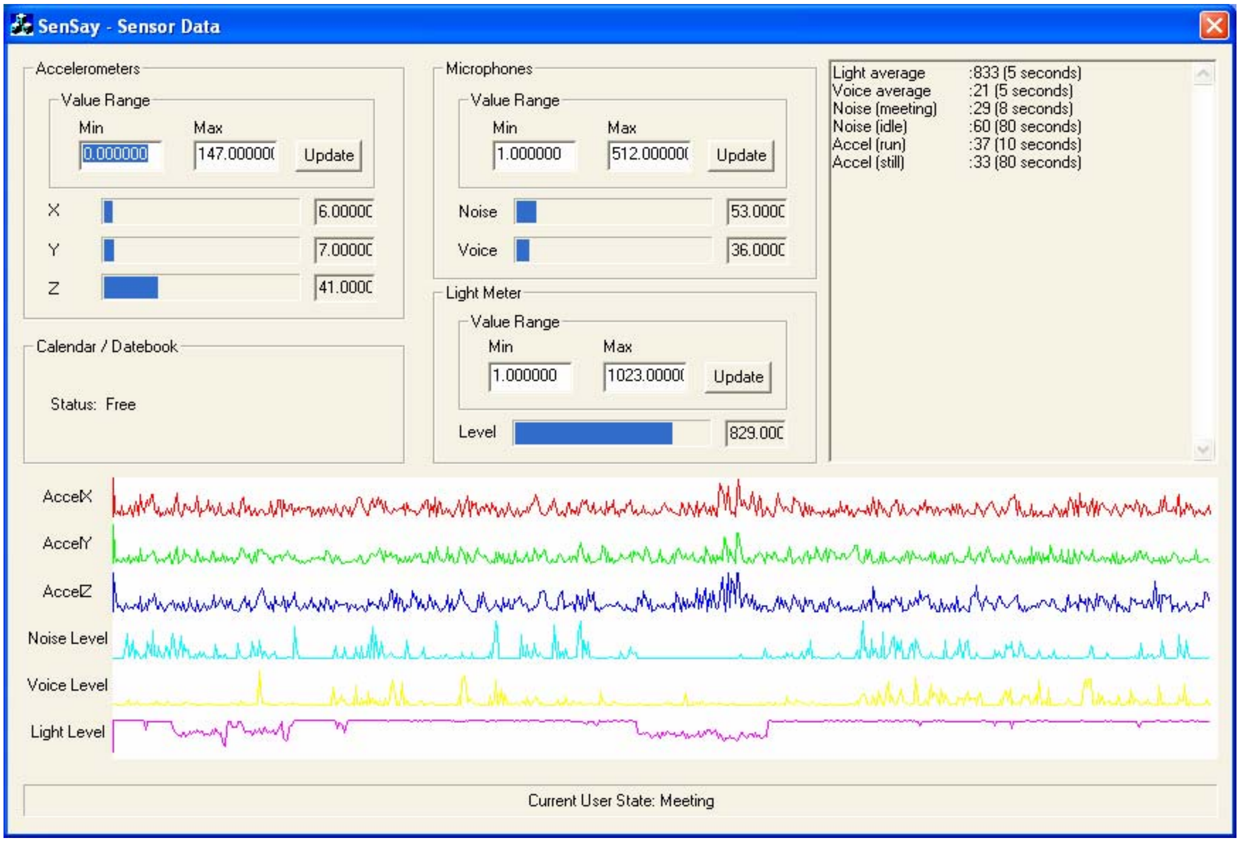

Figure 9. SenSay GUI used to view raw sensor data, averages and trends.

user's pocket or purse, the vibrating feature is turned off.

3.1.2 Active state. SenSay reacts to and facilitates communication despite environmental conditions such as loud music that may keep the user from hearing the ringer. SenSay is capable of detecting high levels of ambient noise and adjusting the ringer and vibrate modes accordingly. If the user is engaged in high-activity physical motion (e.g. jogging or dancing), the phone will dynamically adjust its ringer volume to the loudest setting and turns vibrate on. Before doing this, however, a good deal of physical movement is required before entering this state. It is undesirable that the phone enter the Active state on a quick jog from the office to the lavatory, for example. The decision logic inspects the previous 10 seconds of accelerometer data before determining whether or not the user is engaged in high-activity physical movement.

3.1.3 Idle state. Idle state is defined as a time period when the user is not speaking, is in a relatively low-activity environment, does not have a meeting scheduled, is not speaking very much, and is generally in a 'downtime'. Scenarios for which this state applies are at a bus stop, while eating lunch alone, etc. By definition, if a user is idle, they are interruptible. SenSay assists the user by remembering when a user needs to call a particular person, and prompting the user to place that call during the Idle state. Because these suggestions can be potentially annoying to users, the system inspects all 10 minutes of sensor value history to be highly confident that the user is in the Idle state. Also, the system considers only the previous 5 seconds to determine whether or not to exit the Idle state. These two drastic differences in duration (10 minutes and 5 seconds) creates, in effect, a "difficult to enter", "easy to exit" Idle state. The hope is that the phone will annoy the user only at times where the user is certainly Idle and interruptible.

3.1.4 Normal state. When the decision module does not place the phone in any of the aforementioned states, the phone enters into this default state. In this state, the ringer and vibrate modes are set to the phone's default values and no suggestions are made to the user.

3.2 State Transitions

A Moore finite state machine (FSM) models the phones state. The four states are all connected in the FSM shown in Figure 10, creating a more versatile state machine where the user can transition from any state to any other. A comprehensive list of formal enter and exit criteria was developed for each possible transition. Most transitions are self-explanatory, but two transitions require some formal explanation:

1. Idle to Normal: In the past 30 seconds of the Idle state, if the average of any of the following variables is less than the threshold value for entering into idle mode, exit into normal state.
   - Low motion requirement
   - Low ambient noise requirement
   - Low user voice level requirement
2. Uninterruptible to Normal: This rule applies when the user is in a conversation. If the user hasn't said anything in the past 5 minutes, exit this state and enter the Normal state.

3.3 State Priority

Each state has a number of pre-requisites that must be met before the phone will enter into that state. To address the issue of tie-breakers and relative importance, the rules are prioritized accordingly to subjective measures that will dictate which state to enter into when the environments are similar. For example, if the user is running (which translates into the Active state) while a meeting is scheduled in their electronic calendar at the same time, the phone would enter the Uninterruptible state because a calendar appointment is given a higher priority. The prioritized list of pre-requisites currently consists of:

1. Appointment scheduled in calendar.
2. User is in conversation or speaking aloud.
3. Physical activity is high.
4. Ambient noise is high.
5. Low ambient noise, user activity, and speech.

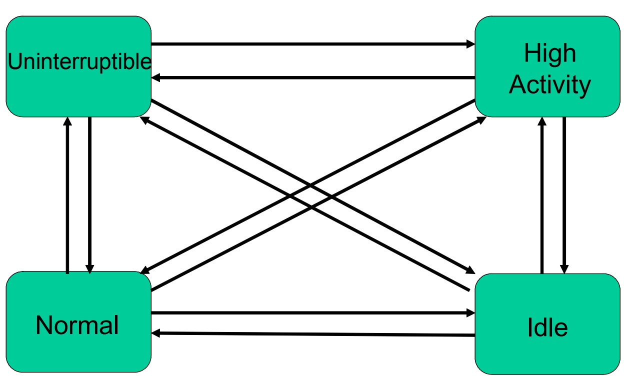

Figure 10. Moore FSM showing all transitions to and from each of the SenSay phone states.

4. Experiment

For each of the possible user states, the need existed to record the corresponding sensor values that pertain to that state. For example, to detect when the user is speaking, baseline readings for a typical conversation were required. A series of threshold analyses tests were ran while recording sensor values and noting trends over time. A special sensor user interface was developed to show numerical and graphical representations of the sensor readings over variable lengths of time. The SenSay Information Visualization software is shown in Figure 9. The history over 10 minutes is shown graphically in the bottom six line graphs and the numerical average over variable time limits on the labels on the top right.

Instead of looking only at the raw sensor values, average sensor values were also observed over various periods of time. For example, to detect a baseline value for a user's voice during conversation, the team observed the average sensor values over a five second time period. The user would speak for five seconds and then watch the running average for another five seconds, bringing the total to ten seconds.

Different sensors warranted different lengths of observation time. This translates loosely to a pseudo-confidence value where the length of time one observes a sensor, the higher the confidence would be. So, if a user's voice was recorded for 5 seconds but a user's high-activity physical motion was recorded for 30 seconds, it means that the phone should be more confident of the user's high activity before transitioning to the Active state. The reason for this is that, again, the phone should decrease the amount of potentially annoying situations as much as possible, while still remaining timely and useful.

The values from each sensor are voltages from 0-5Vdc sampled at 10 bits expressed in integer values from 0-1023. Currently only the raw data values are processed without relating to units such as decibel for sound. This allows us to look at relative changes in the data.

4.1 Test 1: Physical activity

To measure user activity, the sensor board was taped to the user's abdomen area for low noise to ideal mirroring of the user's movements. For the accelerometers, the display shows the changes (deltas) in physical movement instead of the actual movement measurements themselves, because the interesting reading is the amount of gross movement, not the type or direction of that movement.

The physical activity test was run using the 3-axis accelerometer to capture movement. When a user is active one of the accelerometers will show the maximum movement by the user. Therefore in the following measurements, for each sample, the maximum of the three

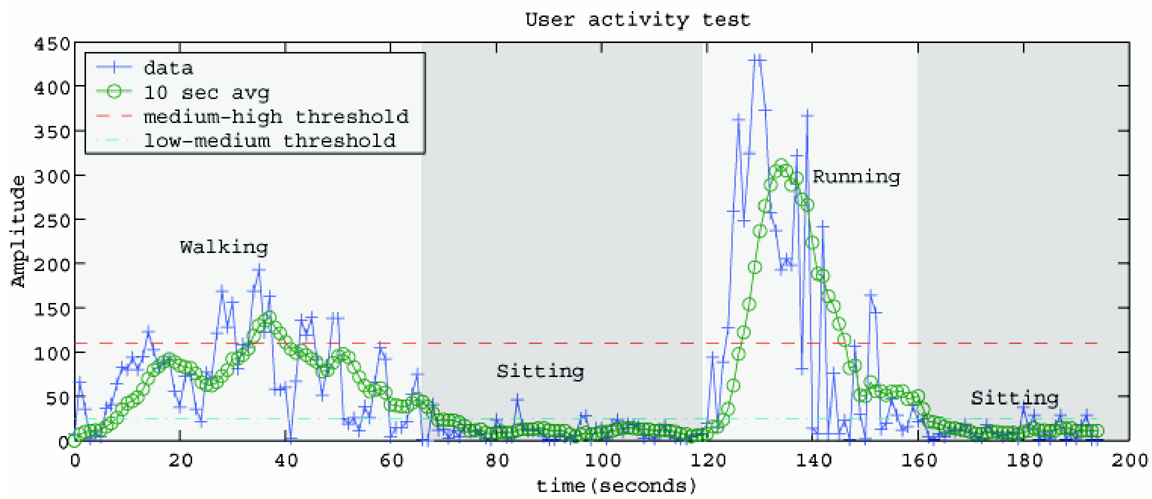

Figure 11. User activity test, broken down into various events.

deltas are used as the movement data. The range of the data is split into three ranges, low, medium and high activity. Each of these states can be seen in the physical activity test shown in Figure 11. The figure includes data from the maximum accelerometer, a 10 second average which is used to smooth out the data, and shows the two threshold values between the three states.

Low Activity represents a user who is sitting, sleeping, etc. anything with a low amount of movement. Movements such as typing, stretching, and other short, bursty movements are averaged out so that the phone does not change states. Low activity is seen in figure 11, in the two areas below the low-medium threshold. The user here was sitting typing at his desk.

Medium Activity represents walking or other comparable activity. Medium activity is represented by the user walking at the beginning of Figure 11. It can be clearly seen that walking provides the expected values, where it can be clearly differentiated from sitting. Medium movement to the platform is when a user is not idle, however no special steps need to be taken to get the user's attention

High Activity represents movements comparable to running. In Figure 11, the user simulated running by jogging in place for 20 seconds. Running causes large spikes in the accelerometer reading that are easily recognizable. High activity means that if a phone call is received, SenSay will need to do everything possible to get the users attention, such as putting the ringer on high and vibrating. The reason for this is when the user is highly active, the user's full concentration will be focused on the activity.

4.2 Test 2: Microphones

The next set of tests was conducted on the two microphones, one capturing user's speech and the other collecting ambient noise data. To capture the user's speech and not get confused with loud surrounding noise, the speech microphone placed on the user's throat, in effect producing a seismometer measuring vibrations of the voice box. The voice microphone is a binary sensor, only needing to differentiate between whether the user is speaking or not.

The ambient noise microphone is an omni-directional microphone mounted on the user's chest facing out to clearly capture surrounding noise. Images of the ambient and voice microphones mounted on the user can be seen in Figure 1. Both the microphones' outputs are rectified so that only positive amplitudes are captured. Also, the data captured from the microphones are averaged, to smooth out spikes in the output.

Figure 12, shows field testing of the microphones over a short period of time with a few staged events. The figure shows the rectified outputs of the microphones, an average of the outputs, and threshold values for each. Different ideal time periods were found previously for averaging the data values. Both the ambient and speech field tests were conducted over the same 300 seconds to show the interaction between the two.

4.2.1 Speech. The first 50 seconds in Figure 12 show high ambient noise with little user speech. Loud music was played to simulate an environment like a club, where the user would have a hard time hearing their phone. It can be seen that the voice microphone behaves ideally, not picking up the surrounding noise. Therefore even at the highest ambient noises, the speech microphone will not cause the

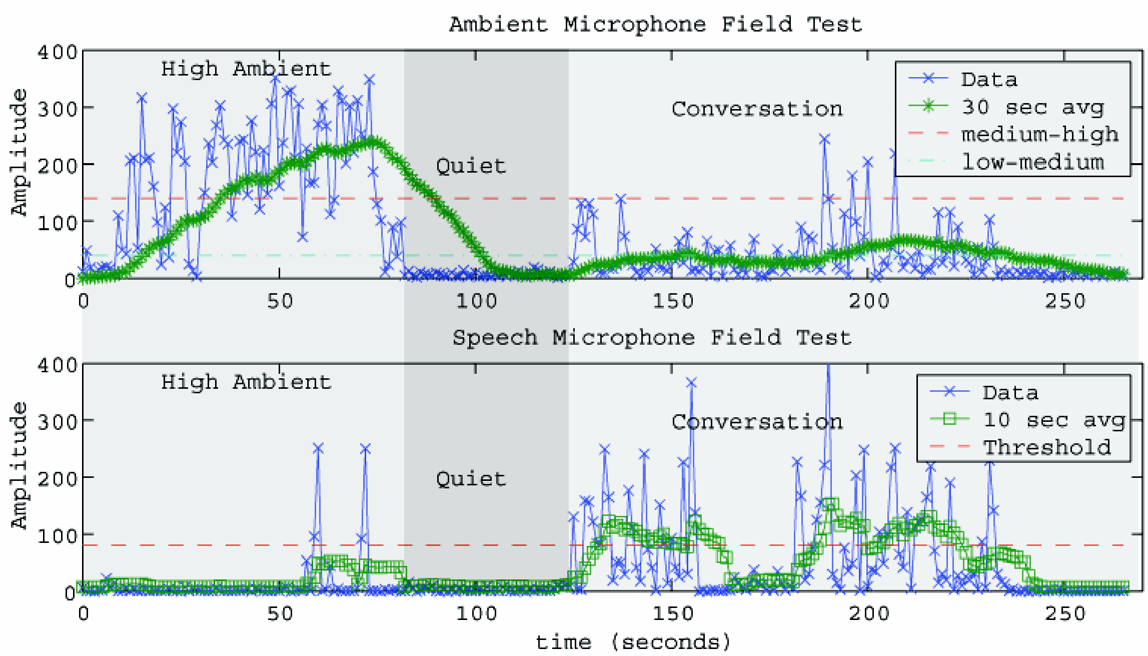

Figure 12. Microphone Field Tests. Both speech and microphones are tested over the same period of time with their data, an average, and their thresholds shown.

phone to change states. The two spikes seen in the high ambient stage are from the user saying a couple of short words, but clearly not in a conversation. The data has large spikes, however, the running average remains below the threshold, allowing the phone to stay in the same state.

The next 50 seconds are spent in a *quiet* state where the user is sitting at his computer typing. The user here made some exaggerated common movements, such as swallowing, to try to skew the data. However, the speech microphone handled these conditions well.

The remaining portions of the example are simulating a conversation between the user and another person. The user's speech causes the microphone to vibrate, and over time the average reaches the threshold to signal the user is in a conversation. One problem the platform addresses is the sections where the speech values drop below the threshold when the user is listening to the response, seen at time 170. The platform assumes that if the user enters into conversation, the user does not exit this state for an extended period. Therefore, momentary drops in the speech values are allowed by the platform.

4.2.2 Ambient Noise. The ambient noise microphone has three states: low, medium and high. High ambient noise is shown in the first 50 seconds of the test as described earlier. High ambient noise is noise relative to that in a club or a bar where the user will have a difficult time hearing the phone. Therefore when in high ambient noise state, the phone turns it's ringer to high and vibrates to try to get the user's attention.

Next, the ambient noise microphone detects low noise. In this test the user was sitting alone at a computer. The ambient noise value drops below the low level. If the user stays in a low ambient noise environment for an extended period, SenSay will go into an Idle state. Also, the ringer volume can be turned to low in these situations.

The last 200 seconds of the test are devoted to a conversation. The ambient noise microphone signals a medium value for this task. Medium noise levels are encountered in many normal states such as conversation, driving a car, or walking outside by traffic. No special actions are taken by the phone when it is in the medium state. This state provides a buffer between the quiet state and the loud state, where special considerations are taken.

4.3 Test 3: Light

The last sensor that was incorporated into the system is the light sensor. The light sensor was attached from the sensor box to the user's phone by a long wire. By placing the light sensor on the phone, SenSay was able to determine weather the phone was in the user's pocket or not. Figure 13 shows a simple test conducted on the light sensor. Again the data values and an average are shown along with the threshold value for the sensor. Throughout the test, the light sensor proved to be extremely effective and sensitive.

At the beginning of the test, the phone was placed in the user's pocket. Next the user removed the phone and placed it on a table. A substantial jump in the output can be seen at this juncture. After another few seconds the user picked up the phone and placed in his hand, to simulate talking over the phone, partially covering the light sensor. The light sensor fluctuates a certain degree, however the average remains over the threshold value throughout the test. Lastly, the phone is returned to the user's pocket, causing the data values to drop. The light sensor is used in various ways by the platform. One use is when the phone is visible to light, it is highly likely that the user is in view of the phone, allowing the phone not to ring and instead just vibrate to get the user's attention.

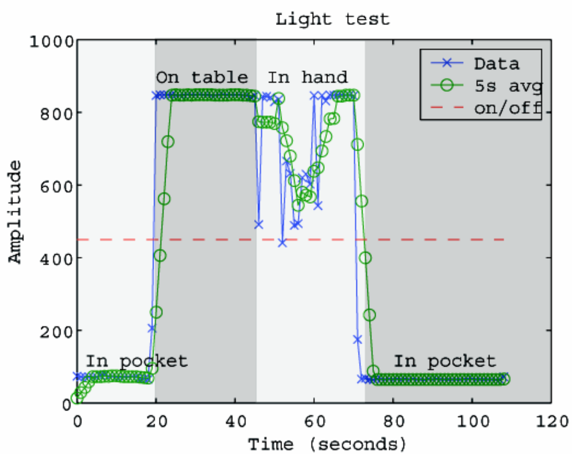

Figure 13. Light sensor test with data, 5 second average and threshold shown.

CONCLUSION

SenSay combines sensory data, user information and history information to create a context-aware phone that provides smart features that aim to improve the overall usability of the cell phone. SenSay can either eliminate unwanted interruptions or actively notify the user of an incoming call by adjusting ringer and vibrate settings. It also has the ability to relay the user's contextual information to the caller when the user is unavailable and it leverages idle time periods by making call suggestions to the user based on call history. We performed threshold analysis on various sensors, which showed that clear delineations can be made between different user states from the data.

In the next version of SenSay, we will first migrate the decision module from a notebook computer to the phone to provide a fully mobile solution. Also SenSay used ideal sensors to provide environmental readings. A major part of the SenSay design was in knowing that the user is in a conversation through the user microphone. However for a fully mobile phone, another method is necessary to retrieve this information.

Acknowledgements

Project advisors were Daniel Sieworick, Asim Smailagic, Susan Finger, and Jeremy Schaeffer. Project participants were James Casazza, Patrick Choi, Mehmet Gerceker, James Grace, Gerard Hamel, Kenneth Herman, C. Kampman Lasater, Anna Li, Daniel Patterson, Seng Tek Sing, Chee Wan Teng, Louis Trebaol, Brandon Weber and Fei Lung Wong.

Reference

[1] A. Schmidt and K.Van Laerhoven. "How to Build Smart Appliances?" *IEEE Personal Communications*, Aug. 2001, pp. 66 - 71.

[2] P. Manalavan, et al. "In-Car Cell Phone Use: Mitigating Risk by Signaling Remote Callers," *ACM Computer Human Interaction Conference,* April 2001.

[3] A. Smailagic and D. Siewiorek. "Application Design for Wearable and Context-Aware Computers," IEEE Pervasive Computing, Vol 1, Num 4, December 2002, pp. 20 – 29.

[4] H. Lieberman and T. Selker, "Out of Context: Computer Systems That Adapt To, and Learn From Context," *IBM Systems Journal,* Nos 3&4, 617632, 2000.

[5] L. Dabbish and R. Baker, "Administrative Assistants as Interruption Mediators," *CHI Conference on Human Factors in Computing*, Pittsburgh, PA, April 2002.