# CenceMe – Injecting Sensing Presence into Social Networking Applications

Emiliano Miluzzo[1], Nicholas D. Lane[1],
Shane B. Eisenman[2], and Andrew T. Campbell[1]

[1] Dartmouth College, Hanover NH 03755, USA
{miluzzo,niclane,campbell}@cs.dartmouth.edu
[2] Columbia University, New York NY 10027, USA
shane@ee.columbia.edu

**Abstract.** We present the design, prototype implementation, and evaluation of CenceMe, a personal sensing system that enables members of social networks to share their *sensing presence* with their buddies in a secure manner. Sensing presence captures a user's status in terms of his activity (e.g., sitting, walking, meeting friends), disposition (e.g., happy, sad, doing OK), habits (e.g., at the gym, coffee shop today, at work) and surroundings (e.g., noisy, hot, bright, high ozone). CenceMe injects sensing presence into popular social networking applications such as Facebook, MySpace, and IM (Skype, Pidgin) allowing for new levels of "connection" and implicit communication (albeit non-verbal) between friends in social networks. The CenceMe system is implemented, in part, as a thin-client on a number of standard and sensor-enabled cell phones and offers a number of services, which can be activated on a per-buddy basis to expose different degrees of a user's sensing presence; these services include, life patterns, my presence, friend feeds, social interaction, significant places, buddy search, buddy beacon, and "above average?"

## 1 Introduction

The growing ubiquity of the Internet provides the opportunity for an unprecedented exchange of information on a global scale. Those with access to this communication substrate, and among these especially the youth, increasingly incorporate information exchange via technologies such as email, blog, instant message, SMS, social network software, and VOIP into their daily routines. For some, the electronic exchange of personal information (e.g., availability, mood) has become a primary means for social interaction [2]. Yet, the question of how to incorporate personal sensing information such as human activity inferencing into these applications has remained largely unexplored. While existing communication forums allow the exchange of text, photos, and video clips, we believe a user experience with a richer texture can be provided in a more natural way by integrating automatic sensing into the various software clients used on mobile

G. Kortuem et al. (Eds.): EuroSSC 2007, LNCS 4793, pp. 1–28, 2007.

communication devices (e.g., cellular phone, PDA, laptop) and popular Internet applications.

It seems we are well situated to realize this vision now. The technology push driving the integration of sensors into everyday consumer devices such as sensor-enabled cell phones is one important enabler, setting the scene for the deployment of large-scale people-centric sensing applications [41] over the next decade. We can observe change in the marketplace: the commonly carried cell phone of today with its microphone and camera sensors is being superseded by smart-phones and PDA devices augmented with accelerometers capable of human activity inferencing, potentially enabling a myriad of new applications in the healthcare, recreational sports, and gaming markets. We imagine people carrying sensor-enabled cell phones or even standard cell phones for that matter will also freely interact over short range radio with other sensors not integrated on the phone but attached to different parts of the body (e.g., running shoes, BlueCel dongle, as discussed in Section 5), carried by someone else (e.g., another user), attached to personal property (e.g., bike, car, ski boot), or embedded in the ecosystem of a town or city (e.g., specialized $CO_2$, pollen sensors).

In this paper, we present the design, prototype implementation, and evaluation of CenceMe, a personal sensing system that enables members of social networks to share their *sensing presence* with their buddies in a secure manner. CenceMe allows for the collection of physical and virtual sensor samples, and the storage, presentation and controlled sharing of inferred human sensing presence. When CenceMe users engage in direct communication (e.g., instant messaging), we aim to allow the conveyance of non-verbal communication that is often lost (or must be actively typed - an unnatural solution) when human interaction is not face to face. When indirect communication is used (e.g., Facebook profile), we aim to make people's personal status and surroundings information - i.e., their sensing presence - available. Similarly, through mining of longer term traces of a user's sensed data CenceMe can extract patterns and features of importance in one's life routine.

The concept of "sensing presence" is fundamental to the CenceMe system, capturing a user's status in terms of his activity (e.g., sitting, walking, meeting friends), disposition (e.g., happy, sad, doing OK), habits (e.g., at the gym, coffee shop today, at work) and surroundings (e.g., noisy, hot, bright, high ozone). CenceMe injects sensing presence into popular social networking applications allowing for new levels of "connection" between friends in social networks. We believe that providing a framework allowing the collection, organization, presentation, and sharing of personal sensing presence and life pattern information will serve a broad spectrum of people, and represents the core challenge of making people-centric sensing [41] a reality. One can imagine many situations where the availability of sensing presence would provide great utility. We have, for example, on the one end a mum who wants to simply know where her kid is, to the youth interested in knowing and catching up with what is hot, where his friends are and what they are doing, what are the trendy hang outs, where a party is taking place, and comparing himself to his peer group and the broader CenceMe community.

The CenceMe architecture includes the following components:

– thin sensing clients focus on gathering information from mobile user communication computing devices including a number of standard and sensor-enabled cell phones. We leverage physical sensors (e.g., accelerometer, camera, microphone) embedded in off-the-shelf mobile user devices (e.g., Nike+ [5], Nokia 5500 Sport [7], Nokia N95 [7]), and virtual software sensors that aim to capture the online life of the user. The thin-client also supports interaction between the phone and external sensors over short-range radio such as the BlueCel dongle (discussed in Section 5), which integrates a 3-axis accelerometer with a Bluetooth radio that can be attached to the body (e.g., as a badge) or other entities (e.g., bike).
– a sensor data analysis engine that infers sensing presence from data.
– a sensor data storage repository supporting any-time access via a per-user web portal (a la Nike+ [5]).
– a services layer that facilitates sharing of sensed data between buddies and and more globally, subject to user-configured privacy policy. CenceMe services can be activated on a per-buddy basis to expose different degrees of a user's sensing presence to different buddies, as needed. These services include *(i) life patterns*, which maintains current and historical data of interest to the user - consider this a sensor version of MyLifeBits [46]; *(ii) my presence*, which reports the current sensing presence of a user including activity, disposition, habits, and surroundings, if available; *(iii) friend feeds*, which provides an event driven feed about selected buddies; *(iv) social interaction*, which uses sensing presence data from all buddies in a group to answer questions such as "who in the buddy list is meeting whom and who is not?"; *(v) significant places*, which represents important places to users that are automatically logged and classified, allowing users to attach labels if needed - in addition, users can "tag" a place as they move around in a user-driven manner; *(vi) health monitoring*, which uses gathered sensing presence data to derive meaningful health information of interest to the user; *(vii) buddy search*, which provides a search service to match users with similar sensing presence profiles, as a means to identify new buddies; *(viii) buddy beacon*; which adapts the buddy search for real-time locally scoped interactions (e.g., in the coffee shop); and finally *(ix) "above average?"*, which compares how a user is doing against statistical data from a user's buddy group or against broader groups of interest (e.g., people that go to the gym, live in Hanover) - in the latter case the presentation of data is always strictly anonymous.
– Consumers of sensing presence run as plugins to popular social networks software (e.g., Skype [9], Gaim/Pidgin [12], Facebook [36], MySpace [37]), rendering a customizable interpretation of the user information.

The rest of the paper is organized as follows. In Section 2, we discuss the CenceMe architecture in more detail, including the types of physical and virtual


**Fig. 1.** Information and process flow in the CenceMe system

sensors we design for, the inference and analysis techniques we use, and the application plugin infrastructure and the per-user web portal that exist to present the distilled sensing presence of the user. We describe the set of services CenceMe provides to applications in Section 3. The CenceMe privacy strategy is discussed in Section 4. Section 5 describes our current prototype implementation; results from the implementation are shown in Section 6. Related work is discussed in Section 7 before we conclude.

## 2 CenceMe Architecture

With the CenceMe system, we aim to leverage the growing integration of sensors into off-the-shelf consumer devices (e.g., cell phones, laptops) to unobtrusively collect information about device users. The goal of the information collection is to allow the system to learn (via data fusion and analysis) characteristics and life patterns of individuals and groups, and to feed this learned information back to users in the form of application services. Noting the increasing popularity of social network applications such as Facebook and MySpace, along with the increasing usage of instant messaging as a replacement to email in both the business world and otherwise, it is clear there is a strong market for the *sharing* of information learned from sensed data as well. We apply this sense/learn/share model in the design of the CenceMe architecture described in the following.

Conceptually, the core of the CenceMe architecture resides on a set of servers that hold a database of users and their sensing presence data, implement a web portal that provides access to processed user data via per-user accounts, and contain algorithms to draw inferences about many objective and subjective aspects of users. APIs to the CenceMe core are used by thin clients running on consumer computing and communication devices such as cell phones, PDAs and laptop computers to push to the core information about the user and his life patterns based on sensed data. While this processed user information is available (both for individual review and group sharing) via the CenceMe web portal, APIs for the retrieval and presentation of (a subset of) this information are used by plugins to popular social network applications (e.g., Skype, Pidgin, Facebook, MySpace) to pull from the core. The CenceMe core in concert with the implemented APIs provide the services discussed in Section 3. A diagram of the relative positioning of the CenceMe core is shown in Figure 2.

In terms of the physical separation of functionality, the CenceMe architecture can be separated into two device classes: back end servers implementing the

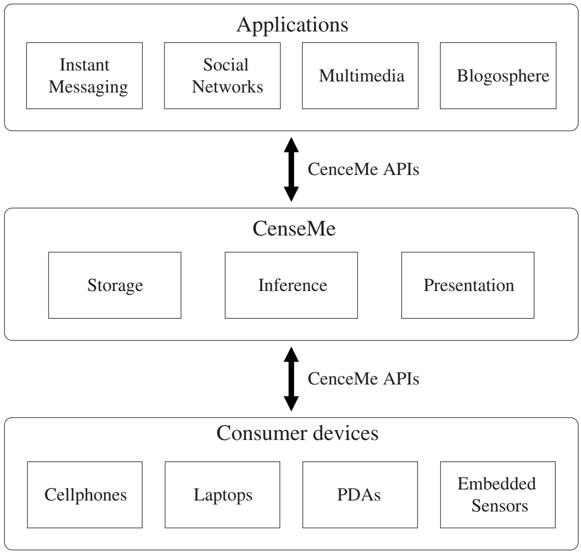



**Fig. 2.** The relative positioning of the CenceMe core between sensors on devices producing data applications consuming information gleaned from sensed data

CenceMe core, and off-the-shelf consumer computing and communications devices that are integrated into the system via APIs to the core. Figure 3 shows the high-level software components that reside on each of these two device classes. Communication between consumer devices takes place according to the availability of the 802.11 and cellular data channels, which is impacted both by the device feature set and by radio coverage. For devices that support multiple communication modes, communication is attempted first using a TCP/IP connection over open 802.11 channels, second using GPRS-enabled bulk or stream transfer, and finally SMS/MMS is used as a fallback. In the following, we describe the sensing, analysis and presentation components in more detail.

### 2.1 Sensing

Conceptually, the thin sensing client installed on the user device periodically polls on-board sensors (both hardware and software) and pushes the collected data samples via an available network connection (wired or wireless) to the CenceMe servers for analysis and storage. For sensing modalities that are particularly resource taxing (especially for mobile devices), sensor sampling may be done on demand via an explicit query. Sampling rates and durations for each of the sensors discussed in this section are set in accordance with the needs of our inferencing engine. Typically, the sensing clients use low rate sampling to save energy and switch to a higher rate sensing upon detection of an interesting *event* (i.e., set of circumstances) to improve sampling resolution. Given the pricing schemes of MMS/SMS and the battery drain implied by 802.11 or cellular

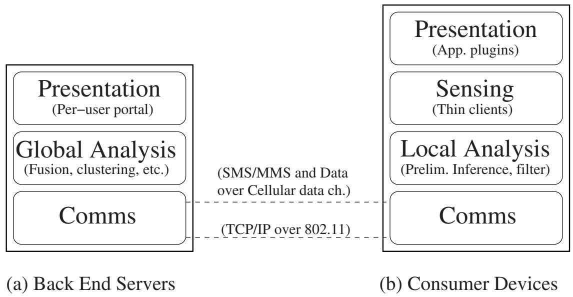



**Fig. 3.** High level Software architecture of the CenceMe core and clients

radio usage, we take further energy-saving and cost-saving measures. Data is compressed before sending to the core, both using standard generic compression techniques on raw data, and domain-specific run-length encoding (e.g., for a stand/walk/run classifier, only send updates to the back end when the state changes). When using SMS, we use the maximum message size to minimize the price per bit. Also, we migrate preliminary data analysis (e.g., filtering, inference) when it makes sense to do so. Given the computational power of most new cellular phones, significant processing can be done on the mobile device to save on communication costs. However, all aggregate (trans-users) analysis is done on the back end. A discussion of the CenceMe hardware and software sensors follows.

**Hardware Sensors.** In the CenceMe architecture, we focus on the following classes of computer communication devices: cell phones like the Nokia N80 and N95 [7]; PDAs like the Nokia N800 [7]; phone/PDA hybrids like Apple iPhone [38]; embedded sensor platforms like Nike+ [5]; recreational sensor platforms like Garmin Edge [40], SkiScape [51] and BikeNet [43]; and laptop/desktop computers. Through a survey of the commonly available commercial hardware, including the examples just mentioned, the following hardware sensors are currently available on one or more COTS devices: embedded cameras, laptop/desktop web cameras, microphone, accelerometer, GPS, radio (e.g., BlueTooth device contact logs, 802.15.4 ranging, 802.11 localization [45] [48]), temperature, light, humidity, magnetometer, button clicks, and device state (e.g., ringer off). CenceMe exploits the availability of these sensors.

**Virtual Software Sensors.** Software sensors are those that measure artifacts of other software that runs on the computing platform in an effort to understand the context of the human's behaviour, mood, etc. They are "virtual" in that they do not sense physical phenomena but rather sense electronic breadcrumbs left as the human goes about his daily routine. Examples of virtual software sensors include, a trace of recent/current URLs loaded by the web browser, a trace of recent songs played on the music player to infer mood or activity, and mobile phone call log mining for structure beyond what your cell phone bill provides.

As an example of how hardware and software sensor samples can be combined to infer activity or status, based on recent web searches (e.g., moviefone.com), and a call to a particular friend, and the time of day, and the day of week, and the phone ringer turned off, we can be nearly sure the human is at the theatre.

### 2.2 Analysis

Sensed data pushed from CenceMe device clients are processed by the analysis component resident on the back end CenceMe servers. Generally, this analysis component combines historical per-user information, with inferences derived from combinations of the current data from multiple sensors to try to reconstruct the status of the individual, i.e., their personal sensing presence. Here we use sensing presence as a broad term meant to cover objective items such as location and activity, and subjective items like mood and preference. While a number of data fusion, aggregation, and data processing methods are possible, the following are examples of analysis/inference outputs we use to generate the sensing presence used by the CenceMe services discussed in Section 3.

Location is a key primitive in any sensing system, to be able to provide geographical context to raw sensor readings. When explicit localization services like GPS are not available either due to hardware limitation or issues with satellite coverage, we infer location of the client devices based on observed WiFi (e.g., access point identifiers), Skyhook service [48], BlueTooth (e.g., static devices) and cellular base station neighborhoods, and other unique sets of sensed data in a manner similar to ambient beacon localization [53].

We incorporate human activity inferring algorithms to log and predict users' behaviour. A simple classifier to determine whether a user is stationary or mobile can be built from several different data inputs, alone or in combination (e.g., changes in location by any possible means, accelerometer data). We evaluate accelerometer data to identify a number of physical activities, including sitting, standing, using mobile phone, walking, running, stair climbing, and others.

Human behaviour is often a product of the environment. To better understand people's behaviour then, it is useful to quantify the environmental context. We gather and process image and sound data to derive the noisiness/brightness of the environment. Further, we run conversation detection and voice detection algorithms to identify the people in a given user's vicinity that may impact behaviour and mood.

Part of a person's daily experience is the environment where the person lives and spends most of the time. For example, health related issues of interest may include the level of an individual's exposure to particulates (e.g., pollen) and pollution. We incorporate mechanisms that enable air quality monitoring around the individual through opportunistic interaction with mobile sensors [43] or static pre-deployed infrastructure [41] [44].

### 2.3 Presentation

Since communication devices, and in particular mobile communication devices, provide varying amounts of application support (e.g., web browser, Skype, and

Rhythmbox on a laptop; web browser and Skype on the N800, SMS only on the Motorola L2 [52]), we provide a variety of means for pulling the human sensing presence distilled from the sensed data from the CenceMe servers and displaying this status on the end user device.

**Text only: Email/SMS.** More limited platforms, such as older/low-end cell phones and PDAs, likely do not have the capability to browse the Internet and have a limited application suite. These platforms can still participate as information consumers in the CenceMe architecture via simple text-based updates via SMS or email, rather than graphical indicators of status embedded in other applications.

**CenceMe Web Portal.** Platforms that support at least general Internet browsing allow users to access their personal CenceMe web portal whose content is served from the CenceMe data repositories. The particular visualizations are customizable to a degree in a manner similar to Google Gadget [59] development/configuration on personalized iGoogle [56] pages. This web portal allows for the most flexible and complete presentation of one's own collected data log, and data shared by others (e.g., via buddy list). Through this portal the user can configure all aspects of his account, including fine grained sharing preferences for his buddies.

**Application-specific Plugins.** Depending on the application support on the device, any of the following plugins are possible. In each case, in addition to status information rendered by the plugin in the applications' GUI, the plugin provides click-through access to the CenceMe portal - both to the user's pages and the shared section of any friends' pages.

– Instant messaging client buddy list shows an icon with a particular status item for the buddy.
– Facebook and MySpace pages have plugins to show your status and that of your friends.
– iGoogle gadgets show various status items from a device user and his buddies. The iGoogle page periodically refreshes itself, so it follows the data pull model from the CenceMe servers.
– Photography applications have plugins to allow pictures to be stamped with metadata like location (minimally) and other environmental (light, temperature) and human status elements.

## 3 CenceMe Services

The goal of the CenceMe system is twofold: (i) to provide information to individuals about their life patterns; and (ii) to provide more texture to interpersonal communication (both direct and indirect) using information derived from hardware and software sensors on user devices. In the following, we describe a number of services built on the CenceMe architecture that aim to meet these goals.

### 3.1 Life Patterns

Enriching the concept put forward in MyLifeBits project [46], we automatically sense and store location traces, inferred activity history [3], history of sensed environment (e.g., sound and light levels), rendezvous with friends and enemies, web search history, phone call history, and VOIP and text messaging history. In this way, we can provide context in the form of sensed data to the myriad other digital observations being collected. Such information may be of archival interest to the individual as a curiosity, and may also be used to help understand behaviour, mood, and health. Sections 5.2, 5.3, and 6 describe our current prototype implementation of the sensing, inferring and display of human activity and environment.

### 3.2 My Presence

As indicated by the increasing popularity of social networking sites like Facebook and MySpace, people (especially the youth) are interested both in actively updating aspects of their own status (i.e., personal sensing presence), and surfing the online profiles of their friends and acquaintances for status updates. However, it is troublesome to have each user manually update more than one or two aspects of his or her sensing presence on a regular basis. We add texture and ease of use to these electronic avatars, by automatically updating each user's social networking profile with information (e.g., "on the phone", "drinking coffee", "jogging at the gym", "at the movies") gleaned from hardware and software sensors.

### 3.3 Friends Feeds

In the same way people subscribe to news feeds or blog updates, and given the regularity with which users of social networking sites browse their friends' profiles, there is clearly a need for a profile subscription service a la RSS (Facebook has a similar service for the data and web interface it maintains). Under this model, friend status updates might be event driven; a user asks to be informed of a particular friends state (e.g., walking, biking, lonely, with people at the coffee shop) at, for example, his cell phone.

### 3.4 Social Interactions

Using voice detection, known device detection (e.g., cell phone Bluetooth MAC address), and life patterns, group meetings and other events that involve groupings of people can be detected. In social group internetworking, friends are often interested in who is spending time with whom. This CenceMe service allows individuals to detect when groups of their friends are meeting, or when illicit rendezvous are happening. A further level of analysis can determine whether a conversation is ongoing (we report results on this in Section 6) and further group dynamics [60] (e.g., who is the dominant speaker).

### 3.5 Significant Places

Have you ever found yourself standing in front of a new restaurant, or wandering in an unfamiliar neighborhood, wanting to know more? A call to 411 is one option, but what you really want are the opinions of your friends. Phone calls to survey each of them are too much of a hassle. Or alternatively, maybe you just want to analyze your own routine to find out where you spend the most time. To satisfy both aims, CenceMe supports the identification and sharing of significant places in people's life patterns.

Significant places are derived through a continuously evolving clustering, classification, and labelling approach. In the first step, we collect location traces from available sources (e.g., wifi association, GPS, etc.) for the given user. Since location traces always have some level of inaccuracy, we cluster the sensed locations according to their geographical proximity. The importance of a cluster is identified by considering time-based inputs such as visitation frequency, dwell time, and regularity. Once significant clusters are identified, a similarity measure is applied to determine how "close" the new cluster is to other significant clusters already identified (across a user's buddies) in the system. If the similarity is greater than a threshold then the system automatically labels (e.g., "Home", "Coffee shop", etc.) the new cluster with the best match. The user has the option to apply a label of his own choosing, if the automatic label is deemed insufficient. Finally, the user has the option of forcing the system to label places considered "insignificant" by the system (e.g., due to not enough visitations yet).

As implied above, the CenceMe system keeps the labels and the cluster information of important clusters for all users, applying them to subsequent cluster learning stages and offering to users a list of possible labels for given clusters. In addition to this "behind the scenes" type of place label sharing, users may also explicitly expose their significant places with their buddies or globally, using the normal methods (e.g., portal, plugins) previously described. In particular, this means that for a user that is visiting a location that is currently not a (significant) cluster to him based on his own location/time traces, the point can be matched against buddies' clusters as well to share. We report on the implementation and performance of this service in Sections 5.2 and 6, respectively.

Once the significant places of users have been automatically identified and either automatically or manually tagged, users may annotate their significant places. The annotation may include identifying the cafe that has good coffee or classifying a neighborhood as either dangerous, safe, hip or dull.

### 3.6 Health Monitoring

As many people are becoming more health-conscious in terms of diet and lifestyle, the CenceMe system also provides individuals with health aspects [1] [34] [21] [33] of their daily routines. CenceMe is able to estimate exposure to ultraviolet light, sunlight (for SAD afflictees) and noise; along with number of steps taken (distance traveled) and number of calories burned. These estimates are derived by combining inference of location and activity [3] of the users with weather information (e.g., UV index, pollen and particulate levels) captured by the CenceMe backend from the web. Inference techniques and results for activity classification and exposure to the weather environment are discussed in Sections 5.2 and 6.

### 3.7 Buddy Search

The past ten years have seen the growth in popularity of online social networks, including chat groups, weblogs, friend networks, and dating websites. However, one hurdle to using such sites is the requirement that users manually input their preferences, characteristics, and the like into the site databases. With CenceMe we provide the means for the automatic collection and sharing of this type of profile information. CenceMe automatically learns and allows users to export information about their favorite haunts, what recreational activities they enjoy, and what kind of lifestyle they are familiar with, along with near real-time personal presence updates sharable via application (e.g., Skype, MySpace) plugins and the CenceMe portal. Further, as many popular IM clients allow to search people by name, location, age, etc., CenceMe enables the search of users through a data mining process that involves also interests (like preferred listened music, significant places, preferred sport, etc).

### 3.8 Buddy Beacon

The buddy search service is adapted to facilitate local interaction as well. In this mode, a user configures the service to provide instant notification to his mobile device if a fellow CenceMe user has a profile with a certain degree of matching attributes (e.g., significant place for both is "Dirt Cowboy coffee shop", both have primarily nocturnal life patterns, similar music or sports interests). All this information is automatically mined via CenceMe sensing clients running on user devices; the user does not have to manually configure his profile information. Devices with this CenceMe service installed periodically broadcast the profile aspects the user is willing to advertise - a Buddy Beacon - via an available short range radio interface (e.g., Bluetooth, 802.15.4, 802.11). When a profile advertisement is received that matches, the user is notified via his mobile device.

### 3.9 "Above Average?"

Everybody is interested in statistics these days. What is popular? How do I measure up? Do I have a comparatively outgoing personality? By analyzing aggregate sensor data collected by its members, CenceMe provides such statistical information on items such as the top ten most common places to visit in a neighborhood, the average time spent at work, and many others. CenceMe makes this aggregate information available to users; each user can configure their portal page to display this system information as desired. Comparisons are available both against global averages and group averages (e.g., a user's friends). Tying in with the Life Patterns service, users can also see how their comparative behaviour attributes change over time (i.e., with the season, semester). The normal

CenceMe privacy model (see Section 4) relies on buddy lists. Therefore, the user must manually opt in to this global sharing of information, even though the data is anonymized through aggregation and averaging before being made available. On the other hand, access to the global average information is only made available to users on a quid pro quo basis (i.e., no free loaders).

## 4 Privacy Protection

Users' raw sensor feeds and inferred information (collectively considered as the user's sensing presence) are securely stored in the CenceMe back end database, but can be shared by CenceMe users according to group membership policies. For example, the data becomes available only to users that are already part of a CenceMe buddy list. CenceMe buddies are defined by the combination of buddy lists imported by registered services (Pidgin, Facebook, etc.), and CenceMe-only buddies can be added based on profile matching, as discussed in Sections 3.7 and 3.8. Thus, we inherit and leverage the work already undertaken by a user when creating his buddy lists and sublists (e.g., in Pidgin, Skype, Facebook) in defining access policies to a user's CenceMe data. Investigation of stronger techniques for the protection of people-centric data is currently underway [11].

Users can decide whether to be visible to other users via the buddy search service (Section 3.7) or via the buddy beacon service (Section 3.8). CenceMe users are given the ability to further apply per-buddy policies to determine the level of data disclosure on per-user, per-group, or global level. We follow the Virtual Walls model [57] which provides different levels of disclosure based on context, enabling access to the complete sensed/inferred data set, a subset of it, or no access at all. For example, a CenceMe user $A$ might allow her buddy $B$ to take pictures from her cell phone while denying camera access to buddy $C$; user $A$ might make her location trace available to both buddies $B$ and $C$. The disclosure policies are set from the user's account control page.

In addition to user-specific data sharing policies, the system computes and shares aggregate statistics across the global CenceMe population. For this service (Section 3.9), shared information is anonymized and averaged, and access to the information is further controlled by a quid pro quo requirement.

## 5 Prototype Implementation

As a proof on concept, we implement a prototype of the CenceMe architecture. Sensing software modules, written both as applications plugins and standalone clients, are installed on commodity hardware, and integrate user devices with the CenceMe core. A sample of the analysis and inference algorithms discussed in the previous sections are implemented as part of sensing clients (preliminary processing) and as back end processes. These automatically process incoming data pushed by the sensing clients. A number of presentation modules are implemented to display information both for individual viewing and sharing between CenceMe users. In the following, we describe the hardware and software details of our prototype implementation.

### 5.1 Sensing

To demonstrate the types of information we can collect from commodity devices and popular applications, we implement a number of sensing clients on a selection of COTS hardware. We use the Nokia 5500 Sport (Symbian OS [42], 3D accelerometer, BlueTooth), the Nokia N80 (Symbian OS, 802.11b/g, BlueTooth), the Nokia N95 (Symbian OS, 802.11b/g, BlueTooth, GPS), the Nokia N800 (Linux OS, 802.11b/g, BlueTooth) and Linux laptop computers. Each sensing client is configured to periodically push its sensed data to the CenceMe core. We provide a short description of each implemented sensing client in the following list.

- Rhythmbox is an open source audio player patterned after Apple iTunes. We write a Perl plugin to Rhythmbox to push the current song to the core. The plugin works on the Linux laptop and the Nokia N800.
- We write a Python script to sample the 3D accelerometer on the Nokia 5500 Sport at a rate that supports accurate activity inference.
- The BlueTooth and 802.11 neighborhoods (MAC addresses) are periodically collected using a Python script. CenceMe users have the option to register the BlueTooth and 802.11 MAC address of their devices with the system. In this way the CenceMe backend can convert MAC addresses into human-friendly neighbor lists.
- We write a Python script to capture camera and microphone samples on the Nokia N80 and Nokia N95 platforms. In addition to the binary image and audio data, we capture and analyze the EXIF image metadata.
- Pidgin is an instant messaging client that supports many commonly used instant messaging protocols (e.g., .NET, Oscar, IRC, XMPP), allowing users to access accounts from many popular IM services (e.g., Jabber, MSN Messenger, AOL IM) via a single interface. We write a Perl plugin to Pidgin to push IM buddy lists and status to the CenceMe core.
- Facebook is a popular web-based social networking application. We write a Perl plugin to Facebook to push Facebook friend lists to the core.
- We write a Python script to periodically sample the GPS location from the Nokia N95.
- Skyhook [48] is a localization system based on 802.11 radio associations. We use Linux libraries compiled for x86 devices and the Nokia N800 to periodically sample the WiFi-derived location and push to the CenceMe core.

In addition to the sensing just described based strictly on commodity hardware, we extend the capability of any BlueTooth enabled device (this includes all the commodity devices we mention above) by allowing for the connection of an external 3D accelerometer. We envision that such a gadget (i.e., a small form-factor BlueTooth/accelerometer accessory) may become popular due to its application flexibility. We implement a prototype BlueCel accessory by integrating a Sparkfun WiTilt module, a Sparkfun LiPo battery charger, and a LiPo battery. The size is 1.5in x 2.0in x 0.5in (see Figure 4). We write a python script to read accelerometer readings from the device over the BlueTooth interface. The placement of the

**Fig. 4.** Mobile devices currently integrated into the CenceMe system: the Nokia N800 Internet Tablet, Nokia N95, Nokia 5500 Sport, Moteiv Tmote Mini (above the N95), and prototype BlueCel accessory (above the 5500)

accessory (e.g., on weight stack, on bike pedal) defines the application. A sensing client menu allows the user to tell the system what the application is, allowing the client to set the appropriate sampling rate of the accelerometer. The data is tagged with the application so that the CenceMe back end can properly interpret the data. Further, we leverage the use of existing embedded sensing systems accessible via IEEE 802.15.4 radio [41] [43] by integrating the SDIO-compatible Moteiv Tmote Mini [10] into the Nokia N800 device.

### 5.2 Analysis

In the CenceMe architecture, unprocessed or semi-processed data is pushed by sensing clients running on user device to the CenceMe core. We implement a MySQL database to store and organize the incoming data, accessible via an API instantiated as a collection of PHP, Perl, and Bash scripts. To extract useful information about CenceMe users from the sensed data, we apply a number of data processing and inferring techniques. We use the WEKA workbench [39] for our clustering and classification needs. In the following, we provide a short description of the data analysis tools we implement in support of the CenceMe services discussed in Section 3. Results on the output of these tools are presented in Section 6.

Based on data from either the Nokia 5500 Sport or the BlueCel accelerometer, we implement an activity classifier (stand/walk/run). The classifier is based on learned features in the raw data trace such as peak and rms frequency, and peak and rms magnitude. This approach applies similar ideas to those found in other accelerometer-based activity inferencing papers (e.g., [62] [3] [14]). This classifier is an example of processing that occurs on the mobile device to avoid the cost (energy and monetary) of sending complete raw accelerometer data via SMS to the back end. Performance results of this classifier are shown in Section 6.1.

We construct a classifier for determining whether a user is indoors or outdoors. We combine a number of elements into the feature vector to be robust to different types of indoor and outdoor environments. The features we consider are: the ability of the mobile device to acquire a GPS estimate, number of satellites seen by GPS, number of WiFi access points and BlueTooth devices seen and their signal strengths, the frequency of the light (looking for the AC-induced flicker), and differential between the temperature measured by the device and the temperature read via a weather information feed (to detect air conditioning). Performance results of this classifier are shown in Section 6.1.

We construct a mobility classifier (stationary/walking/driving) based on changes to the radio neighbor set and the relative signal strengths (both for individual neighbors and the aggregate across all neighbors), for BlueTooth, WiFi, and GSM radios, respectively, of the mobile devices. The idea is to map changes in the radio environment (i.e., neighbors, received signal strength) to speed of movement. The classifier uses techniques similar to those used in existing work [54] [55] [50]. The result of the aforementioned indoor/outdoor classifier is also included in the feature vector. Locations traces are omitted due to their relatively high error with respect to the speed of human motion. Performance results of this classifier are shown in Section 6.1.

Using Matlab processing on the back end, we generate a noise index (expressed in decibels) from audio samples captured from N80 and N95 microphones. Similarly, using Matlab we generate a brightness index (ranging from 0 to 1) from images captures from N80 and N95 cameras. The sounds and brightness indices help us to infer information about a person's surroundings. In particular, we combine the noise index to estimate the cumulative effect of the sound environment on a user's hearing, and the positive effect of sunlight (when combined with an indoor/outdoor classifier) on those afflicted with seasonal affective disorder. Finally, we implement a classifier based on a voice detection algorithm [19] to determine if a user is engaged in a conversation or not. Performance results of this classifier are shown in Section 6.1.

By analyzing location traces, time statistics of mobility, and other data inputs, as described in Section 3.5 we derive a user's significant places. Raw location data is first clustered using the EM algorithm, then clusters are mapped against time statistics (viz., visitation frequency, dwell time, regularity, time of day, weekday/weekend, AM/PM) and other information (viz., indoor/outdoor, current and previous mobility class, number and composition of people groups visible in a location) to determine importance. Also, WiFi and Bluetooth MAC address of neighbors are used to differentiate between overlapping clusters. Finally, a similarity measure is computed between the new cluster and existing clusters known by the system. The system maintains generic labels for these significant clusters, but users may alias them as well to give more personally meaningful or group-oriented names. The clustering is adaptive since the model changes over time depending on how the mobility trace of the user (and other system users) evolves (the significance of a place may evolve over time). The algorithm to recognize significant locations by combining location trace data with other indicators shares concepts with prior work [49] [47] [66] [65] [63]. The CenceMe approach is distinguished by the way it clusters according to per-user models (rather than globally), and then shares models based on social connections (e.g., presence in a buddy list). This provides a more personal/group-oriented set of labeled significant places, at the expense of general applicability of the training data. While it is often advantageous to relate recognized significant clusters to physical locations (i.e., coordinates), we also enable the recognition of significant places for devices that do not have access to absolute localization capabilities with the use of local region recognition based on what is available to the device (e.g., WiFi, Bluetooth, GSM) [61]. In this way, a location cluster does not solely have to be an aggregated set of true coordinate estimates, but can comprise a set of location recognition estimates, a la ABL [53].

In terms of health and fitness, we estimate the number of calories burned by combining the inference of walking from the standing/walking/running classifier, time spent walking, and an average factor for calories burned per unit time when walking at a moderate pace [20]. Further, we estimate exposure to ultraviolet light by combining the inference of walking or running or standing, the inference of being outdoors, the time spent, and a feed to a web-based weather service to learn the current UV dose rate [24]. A similar technique is applied to estimate pollen exposure (tree, grass, weed) and particulate exposure.

As discussed in Section 5.1, the BlueCel facilitates a number of application-specific data collection possibilities. We implement commensurate application-specific data analysis tools for bicycle rides (BlueCel placed on the pedal), golf swing analysis (BlueCel affixed to the club head), and analysis of weight lifting activity for exercise motion correctness (injury avoidance) and workout logging (BlueCel affixed to the wrist).

### 5.3 Presentation

All of a user's processed sensor data can be viewed via a web browser by logging into the user's account on the CenceMe portal. Additionally, a subset of the user's status information is made available (via both data push and data pull mechanisms) to the user's buddies (subject to his configured sharing policies) through their CenceMe portal pages, and through plugins to popular social networking applications.

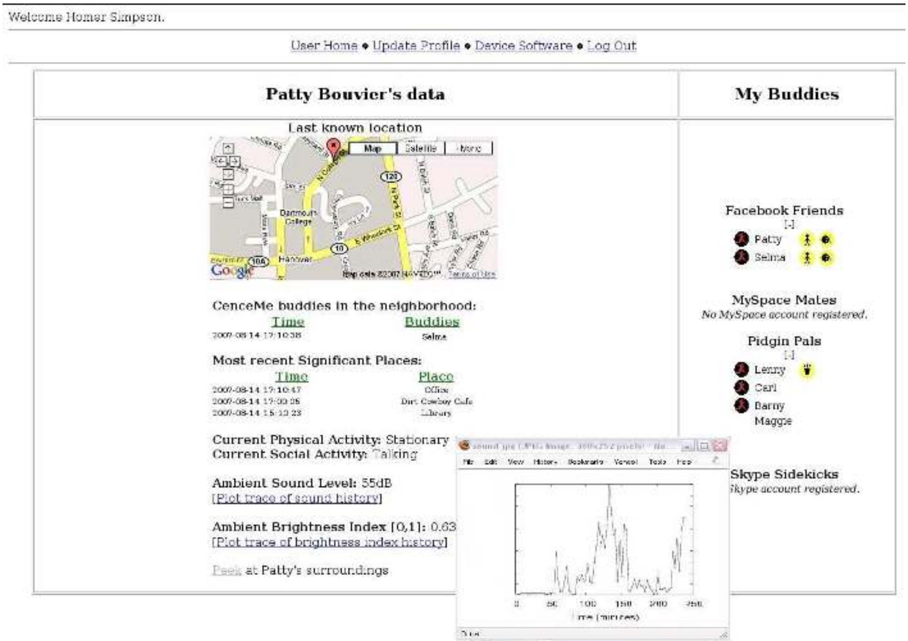



**Fig. 5.** Portal snaphot

Typically, the data a user shares with his buddies is rendered via a number of simple icons that distill the current sensing presence of the user. Figure 5 shows a snapshot of the data page of user on the CenceMe portal we implement. Buddy lists loaded on the right pane from registered Pidgin and Facebook accounts are annotated with icons representing the shared data. The icons offer click-through access to a fuller representation of the shared user data. In the Figure, buddies Patty and Selma are inferred to be standing and in a conversation, while buddy Lenny is inferred to be at the coffee shop, as indicated by the yellow activity icons next the each buddy's name. On login, the left pane shows the data of the logged in user, but shows a buddy's data if any one of the icons next to that buddy's name is clicked. In this case, the logged in user Homer Simpson has clicked on the icon for his buddy Patty. Patty has a sharing policy that allows the display of the data shown in the left pane: Patty's buddies in her vicinity (via BlueTooth and WiFi MAC address recognition), Patty's trace of her last significant places visited, etc. In sum, this is the detailed data behind the iconic representation of "standing" and "talking". Note that the link at the bottom of the page to take a picture ("peek") from Patty's cell phone is disabled; Patty has disabled access for Homer Simpson to image data in her privacy profile. Instead, Homer has clicked the link to view the sound level history plot for Patty, ostensibly to see how noisy Selma is. The black icons denote that a buddy imported from the user's Facebook account (using the Facebook developer API) or Pidgin account (using the Pidgin developer API) is also a registered CenceMe user.

We also implement application plugins for Pidgin and Facebook that offer click-through access to the buddies data page via a web browser launched on the mobile device. As part of ongoing work, we are extending the presentation capabilities of CenceMe to include application plugins for Skype and MySpace. Additionally, we envision sensing and presentation plugins will be written by interested third parties, using the provided CenceMe APIs.

## 6 Results

While our system implementation of the CenceMe architecture is still under development, in the following we present selected results on aspects of the current system performance, and give a flavor for some of the applications that the CenceMe system supports.

### 6.1 Classifier Performance

The CenceMe services outlined in Section 3 strongly rely on the ability of the analysis components (running both on the mobile devices and the back end servers) to glean meaningful insights about a user's life from a possible myriad of sensed data feeds. To provide a measure of the initial quality of the CenceMe services we provide, we present the performance of a number of classifiers we implement. Implementation descriptions of the presented classifiers are given in Section 5.2.

**Activity: Standing/Walking/Running.** Our activity classifier is currently based exclusively on accelerometer data. Since this sensor may not always be available on all mobile devices, beyond this initial evaluation the classifier will incorporate a broader array of sensors to improve the robustness of the technique. Additionally, we are in the process of enhancing the activity classifier to detect a broader range of activities.

We run experiments to evaluate the accuracy of the mobility classifier described in Section 5.2. The accelerometer thresholds of the classifier (i.e., between stationary/walking and walking/running) are learned based on mobility traces from three different people. The results shown in Figure 6(a), give the average test results for two others. The accelerometer is tested in two mounting positions (viz., belt, pocket) and the values represent the average of 4 one hour experiments. The scenarios include a normal office setting behaviour and sports (walking/running to the campus Green and back). From the matrix in Figure 6(a) we see that activities are classified correctly approximately 90% of the time for each of the three activities.

**Mobility: Stationary/Walking/Driving.** Our evaluation of the mobility classifier (described in Section 5.2) results from a week long controlled experiment with four members in our lab. All members carry Nokia N95 cell phones that execute the classifier. Participants manually label any state changes, which provide both training data for the classifier and a data set from which classifier

| | Standing | Walking | Running |
|---|---|---|---|
| Standing | 0.9844 | 0.0141 | 0.0014 |
| Walking | 0.0558 | 0.8603 | 0.0837 |
| Running | 0.0363 | 0.0545 | 0.9090 |

(a) Activity classifier.

| | Stationary | Walking | Driving |
|---|---|---|---|
| Stationary | 0.8563 | 0.3274 | 0.1083 |
| Walking | 0.1201 | 0.6112 | 0.2167 |
| Driving | 0.0236 | 0.0614 | 0.6750 |

(b) Mobility classifier.

| | Indoors | Outdoors |
|---|---|---|
| Indoors | 0.9029 | 0.2165 |
| Outdoors | 0.0971 | 0.7835 |

(c) Indoor/outdoor classifier.

| | Background noise | Conversation |
|---|---|---|
| Background noise | 0.7813 | 0.1562 |
| Conversation | 0.2187 | 0.8438 |

(d) Conversation classifier.

**Fig. 6.** Confusion matrices for the implemented classifiers

performance is derived. Figure 6(b) shows the confusion matrix that results with one third of the data used for training and the remainder used for evaluation. From this figure we observe that walking and driving are less accurately labeled. This is because the difference between these two states is less distinct. We intend to refine the classifier to incorporate additional modalities (such as audio) to address this weakness. We perform ten fold cross validation on our experiment data and find that 81% of instances still have correct labels. This allays some concerns about potential over-fitting.

**Indoor/Outdoor.** The evaluation of the indoor/outdoor classifier (described in Section 5.2) follows a similar methodology to that of the mobility classifier, with participants manually labeling their state changes. Since the results of the classification trials that we present here are performed only on the N95 hardware, many of the vector features that are part of the process have no effect (e.g., detection of the light flicker is not possible with this hardware). We perform independent tests with alternative sensing hardware, the Tmote Sky [10], using only temperature, light and humidity sensors available and find approximately 83% classification accuracy is achievable. As the sensing capabilities of cell phones mature, we will evolve our classification techniques to take advantage of the commercial technology. The results shown in Figure 6(c) are very promising given the initial stages of the development of this classifier. We observe that an accuracy of 86% results when we perform 10 fold cross validation.

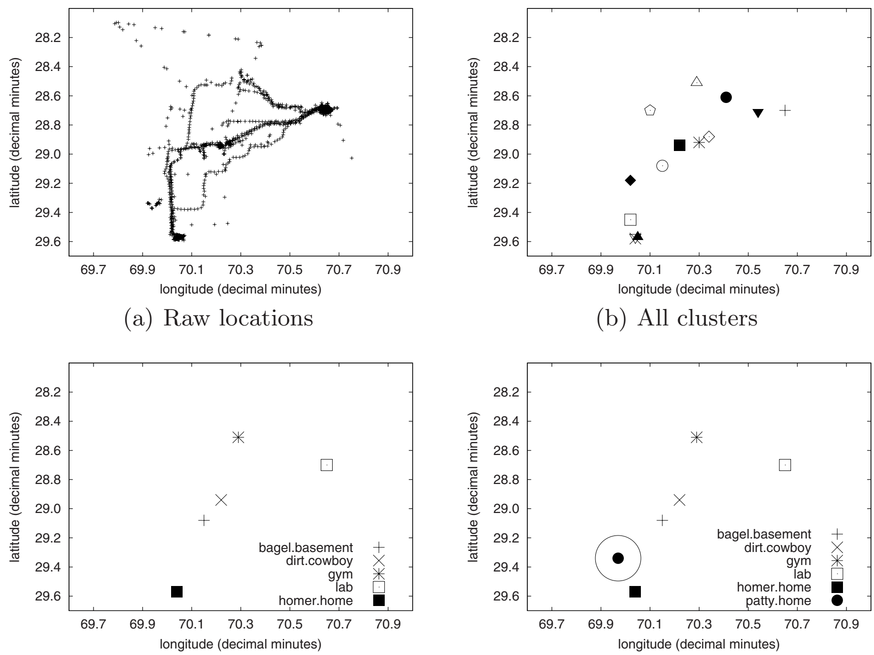
We must qualify these results by saying these were performed in a college campus environment with a rather dense WiFi AP deployment. Further there are no significant high rise buildings that may induce urban canyoning effects with the GPS signal. On the other hand, due to the low population density of the area there is also a lower density of cell phone towers. As a result, some feature vector elements that are important within our test region will not be important for all regions. We plan to perform tests as part of a broader system study to evaluate the classifier performance in substantively different environments from that used for these initial tests.

**Conversation detection.** We conduct experiments to evaluate the accuracy of the conversation detection algorithm [19]. The experiment consists of taking samples from the cell phone's microphone from inside a pocket (to reproduce the common scenario of cell phones carried by a person) during conversations between a group of people in an office setting, sidewalk, and a restaurant. We annotate the time of the conversation for ground truth purposes. The microphone samples are also time stamped for comparison with the ground truth data. The result of the conversation detection algorithm is reported in Figure 6(d). The background noise and an ongoing conversation are correctly detected with an accuracy of almost 80% and 84% respectively. On average, 15% and 21% of the samples are mistaken, respectively, for conversation when they are noise and for noise when they are conversation. We conjecture that we can decrease the error by further processing the sound samples by applying low pass filters. In this way, sounds characterized by frequencies higher than 4KH (e.g, background noise) could be filtered out allowing a more accurate voice detection. As part of future work, we will augment our system with stronger voice processing capabilities. With these new capabilities, CenceMe users will have the ability to know who is involved in a conversation (subject to user privacy policy), rather than just a binary classification.

### 6.2 Significant Places

For the purpose of evaluating the CenceMe significant places service, four members of our lab execute data collection client code on Nokia N95 cell phones that sample and construct the types of data features discussed in Section 3.5. In the following, we demonstrate the sharing of significant place models (as described in Section 5.2) between buddies in the context of the general operational flow of the service.

For the purposes of this scenario we examine the interaction between two of the four system users. In Figure 7(a), we observe the location trace for a single user, Homer, with his location trace collected over the course of a week around Hanover, NH. Figure 7(b) provides the result of basic EM clustering of these raw location estimates. Figure 7(c) provides the results of the significant places process based only on Homer's data. These are visitations to cluster instances which are classified by the system as being significant. The process begins with a default classifier based on training data sourced from the user population. The classifier is then refined by input from Homer. This figure shows these cluster visits having semantic labels. In this case these labels are solely provided by Homer himself. In Figure 7(d), a new significant location appears. This location represents the home of another user, Patty. Homer visited Patty's home and a raw cluster is created by this visitation (as seen in Figure 7(b)). However, since Homer had never visited this location before, although it is recognized as being significant a label can not be determined. Instead, to label this cluster the system executes Homer's buddy Patty's models on the data collected by Homer. Given a sufficiently good match, the labeling is performed and appears in visualizations such as Homer's log of his sensing presence accessible via the CenceMe portal. Importantly, the ability to recognize and label such visitations is then incorporated into Homer's significant place model.




(a) Raw locations

(b) All clusters

(c) Significant clusters based on single user's data

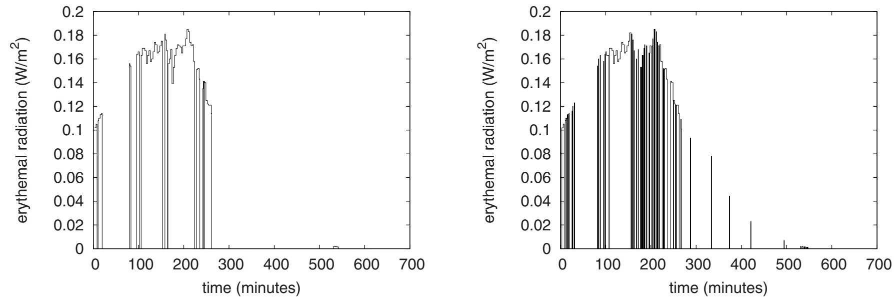
(d) Additional labeled cluster added by cross-referencing against buddy's labeled significant clusters.

**Fig. 7.**

## 6.3 Health

By combining the output of the indoor/outdoor classifier and the mobility classifier with an external weather feed that provides a trace of the UV dose rate (i.e., erythemal irradiance) [24], we estimate a person's exposure to harmful UV radiation. This combination of classifiers tells us when a person is outside a building but not inside a motor vehicle. Figure 8(a) shows the estimated UV exposure of the person if indoor/outdoor classification is perfect, while 8(b) shows the estimated UV exposure given the actual classifier performance. The figures are in excellent agreement, underscoring the good performance of the classifier combination. Further, in presenting this result we show how simple classifiers can contribute to monitoring important human health features. As future work we will compare the estimated results against real measured exposure to further improve the classifier combination we use.

Similarly, we use the output of the mobility classifier to estimate how many calories a person is burning. In Figure 9, we show a comparison between the estimated cumulative calories burned using the classifier output and the actual activity, respectively. The plots reflect a real seven hour human activity trace. In both cases, the time spent walking, either inferred from the classifier or taken




(a) Estimated UV exposure based on perfect classification.

(b) Estimated UV exposure based on actual classification.

**Fig. 8.**

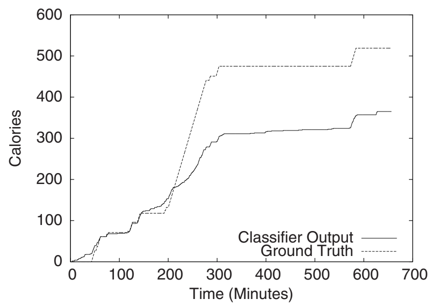


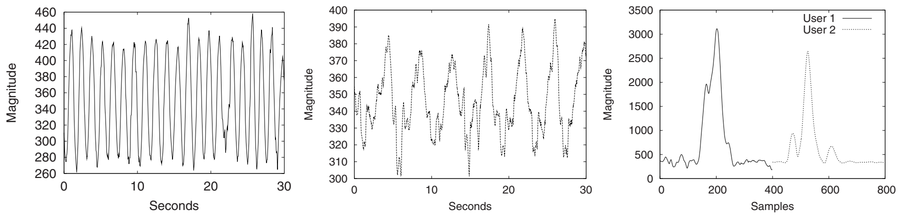

**Fig. 9.** A comparison of the estimated calories burned based on the ideal and actual performance, respectively, the mobility classifier

from the ground truth log, is multiplied by a standard average factor for calories burned while walking [20] to calculate the estimate.

### 6.4 BlueCel Applications

The flexibility of the BlueCel accessory sensor allows people to run many different applications that are of interest to them, with a single multi-purpose device. This external, Bluetooth-connected accelerometer offers advantages even over accelerometers that are integrated into mobile phones (e.g., Nokia 5500 Sport) since for many applications it is required or at least convenient that the accelerometer be in a different place than mobile phones are normally carried (e.g., pocket, hand bag). Further, the form factor of a mobile phone is too large to facilitate useful data collection for some applications.

To demonstrate the flexibility of the BlueCel approach, we implement and collect data from three simple and diverse applications. For each of these applications, the accelerometer signature in terms of combined three channel




(a) Bicycling. (b) Weight lifting. (c) Golf swing analysis.

**Fig. 10.** Use of the BlueCel sensor to add a 3d accelerometer to any BlueTooth-equipped mobile device supports a number of applications. The plots in (a), (b) and (c) show BlueCel signatures, the combined magnitude of the three channel output of the accelerometer. A moving average with a window size of 25 samples is applied to the raw accelerometer samples for all three plots.

magnitude (i.e., $\sqrt{x^2 + y^2 + z^2}$) is plotted; sampling is at 37 Hz. Figure 10(a) shows a 30 second excerpt from a bicycle pedaling analysis experiment. The BlueCel is slipped inside a rider's sock. A user can easily do the same to determine his pedaling cadence, for training purposes or just for fun. Figure 10(b) shows the accelerometer signature for several repetitions of a weight lifting exercise. Though also periodic, the signature is quite distinct from that of the pedaling. We can easily log workout statistics by analyzing this signature [6]. As future work, with further processing of the signature we hope to provide an indication of whether the exercise is being performed properly in terms of range of motion. Figure 10(c) shows the golf swing signatures of two different users, one a novice and the other more experienced. The signatures are quite distinct, that of the experienced golfer (User 1) being more smooth and compact. A novice user might use such comparisons as a guide to iteratively analyze and modify his swing for improvement. These three are just a sample of possible applications, and users themselves have the freedom (due to the flexibility of the BlueCel model) to come up with their own applications.

## 7 Related Work

Much attention has been paid in the research community to the intersection of social networking and communication technology. In particular, cell phones have long been recognized as an ideal platform for pervasive applications (e.g., [58]). They are increasingly seen as a platform of choice for urban and people-centric sensing systems [32] [29] [31] [15] [35] [30] [4]. They are well suited for this domain due to their ubiquity, expanding suite of sensors and ability to interact with additional external sensors via short range radio. Further, given the increasing market penetration of cellular phones and the parallel trend of sole reliance on cell phones for telephonic service, they are likely to be carried at all times. Data collection and sharing via cell phones and similar mobile devices are key enablers of the CenceMe architecture. In the following, we summarize exisiting research incorporating cell phones and other commodity devices into the data collection and sharing architecture.

Mobile devices like cell phones have been used to learn about social connections, an important point of the CenseMe service architecture. Contributions in this domain include the MIT Reality Mining project. In [26], the authors collect BlueTooth neighbor data from one hundred mobile phones over a period of nine months in order to identify social connections between people and patterns in their daily life. Follow on work explores the notion of sampling social connections with data such as proximity and sounds [18], and applies principal component analysis to location/time series data and BlueTooth neighborhoods to learn social groups [25].

One use of data collected by cell phones is to facilitate context-aware communications, wherein availability data is collected from personal devices and shared with friends. In [8] the authors describe iCAM, a web-based address book for cellular phones. System users opt-in to share communication contexts via a web interface to expose the preferred method of communication at a given time (e.g., in person, email, home/work phone). Contexts are generated by cellular tower-based localization and manually configured schedule information. Registered rules govern the type of information exposed. The authors of [13] propose and build a system similarly aimed at choosing the best communication modality to interact with close friends and family. The system uses GPS location, accelerometer data to determine between walking and driving, and a microphone to determine between talking and silent. CenceMe goes beyond both [8] and [13] by taking as inputs a broader set of sensor feeds, learning patterns in each user's life automatically rather than relying on manually input schedule information, and outputting status information much richer than just current location and communication preference. Additionally, CenceMe is not limited to running on customized hardware (e.g., the PHS in [8], the WatchMe watch in [13]), but is integrated into popular social applications (using supported APIs) already running on commodity hardware (cellular phone, laptop computer, PDA, etc.).

The CenceMe notion of sensing presence sharing and logging differs from the idea of presence sharing and exchange presented in [22] and [23]. The latter idea refers to the beaconing of personal information using short range radio. The CenceMe notion incorporates in situ exchanges (e.g., via the buddy beacon service), but extends beyond this simple interaction. CenceMe is focused on the process of distilling the sensed presence of the individual from COTS devices and sharing the sensed presence irrespective of the actual proximity between users, largely based on social groups (e.g., buddies) via existing applications. Twitter [28] is more closely aligned with CenceMe in terms of sharing personal status information, but is typically limited to manually generated text-based status sharing. The primary benefit of [28] is the ability to aggregate and distribute these status messages from and to multiple points (i.e., cell phones, IM and the Web). CenceMe extends beyond sharing text messages and focuses on the automated distillation of users' personal sensed presence. The work presented in [17] provides a limited exploration of the sharing sensing presence between socially connected parties. This work shares simple moving/not moving status while investigating the utility and privacy concerns of sharing such status. As an indication of the demand for a more capable system, participants of the study requested that richer forms of presence sharing be offered [17].

Alternative proposed architectures that enable people-centric sensing often rely on specialize hardware, rather than commodity consumer devices like cell phones. The architecture proposed in [27] relies on users to attach numerous cheap radio tags to everyday objects with which they come in contact. Through proximity detection between user devices and object tags, activity recognition is possible. The Mithril project [64] is representative of those assuming more capable devices and is built around a linux PDA with multiple body-attached sensors. Similarly, the SATIRE project [16] builds a system with greater levels of sensing capability using "smart clothing". With SATIRE, networked mote-class devices are embedded in clothing such as a jacket. CenceMe uses heterogeneous COTS devices (e.g., cell phones) already in widespread use. CenceMe collects data from the sensors available on these devices (e.g., BlueTooth, GPS, accelerometer, camera, microphone), and in support of services and applications not possible with proximity data alone. The CenceMe architecture leverages the idea of integrating simple external sensors devices (e.g., the BlueCel device, c.f. Section 5.1) as application-specific add-ons to the cell phone. This model is similar to [30] and [14], both of which assume additional expansion boards are attached to a standard cell phone.

## 8 Conclusion

We have presented a detailed description of the CenceMe architecture. Through our prototype implementation we have demonstrated successful integration with a number of popular off-the-shelf consumer computer communication devices and social networking applications.

## Acknowledgment

This work is supported in part by Intel Corp., Nokia, NSF NCS- 0631289, and the Institute for Security Technology Studies (ISTS) at Dartmouth College. The authors thank Hong Lu for his help with data collection, Ronald Peterson for building the BlueCel, and Peter Boda and Chieh-Yih Wan for their support of this project. ISTS support is provided by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

## References


1. Lester, J., et al.: Sensing and Modeling Activities to Support Physical Fitness. In: Proc. of Ubicomp Workshop: Monitoring, Measuring, and Motivating Excerise: Ubiquitous Computing to Support Fitness, Tokoyo (September 2005)
2. Kids say e-mail is, like, soooo dead. CNET (July 19, 2007), http://news.com.com/Kids+say+e-mail+is,+like,+soooo+dead/2009-1032_3-6197242.html
3. Lester, J., et al.: A Hybrid Discriminative/generative Approach for Modeling Human Activities. In: Proc. of the 19th IntłJoint Conf. on Artificial Intelligence, Edinburgh, pp. 766–722 (2005)
4. MetroSense Project Web Page, http://metrosense.cs.dartmouth.edu
5. Nike$^{+}$, http://www.nikeplus.com
6. Chaudhri, R., et al.: Mobile Device-Centric Exercise Monitoring with an External Sensor Population (Poster abstract). In: Proc of 3rd IEEE Int'l. Conf. on Distributed Computing in Sensor Systems, Santa Fe (June 2007)
7. Nokia, http://www.nokia.com
8. Nakanishi, Y., Takahashi, K., Tsuji, T., Hakozaki, K.: iCAMS: A Mobile Communication Tool Using Location and Schedule Information. IEEE Pervasive Computing 3(1), 82–88 (2004)
9. Skype, http://www.skype.com
10. Moteiv Tmote Mini, http://www.moteiv.com/
11. Johnson, P., Kapadia, A., Kotz, D., Triandopoulos, N.: People-Centric Urban Sensing: Security Challenges for the New Paradigm. Dartmouth Technical Report TR2007-586 (February 2007)
12. Gaim/Pidgin, http://sourceforge.net/projects/pidgin
13. Marmasse, N., Schmandt, C., Spectre, D.: WatchMe: Communication and Awareness Between Members of a Closely-knit Group. In: Proc. of 6th Int'l. Conf. on Ubiq. Comp., Nottingham, pp. 214–231 (September 2004)
14. Lester, J., Choudhury, T., Borriello, G.: A Practical Approach to Recognizing Physical Activities. In: Proc. of 4th Int'l. Conf. on Perv. Comp., Dublin, pp. 1–16 (May 2006)
15. Abdelzaher, T., et al.: Mobiscopes for Human Spaces. IEEE Perv. Comp. 6(2), 20–29 (2007)
16. Ganti, R., Jayachandran, P., Abdelzaher, T., Stankovic, J.: SATIRE: A Software Architecture for Smart AtTIRE. In: Proc. of 4th. Int'l. Conf. on Mobile Systems, Applications, and Services, Uppsala (June 2006)
17. Bentley, F., Metcalf, C.: Sharing Motion Information with Close Family and Friends. In: Proc of SIGCHI Conf. on Human Factors in Computing Systems, San Jose, pp. 1361–1370 (2007)
18. Eagle, N., Pentland, A., Lazer, D.: Inferring Social Network Structure using Mobile Phone Data (in submission, 2007)
19. Matlab Library for Speaker Identification using Cepstrul Coeff., http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8802
20. Food, Nutrition, Saftey and Cooking Site at the University of Nebraska - Lincoln, http://lancaster.unl.edu/food/
21. Borriello, G., Brunette, W., Lester, J., Powledge, P., Rea, A.: An Ecosystem of Platforms to Support Sensors for Personal Fitness. In: Proc of 4th Int'l. Workshop on Wearable and Implantable Body Sensor Networks, Aachen (March 2006)
22. Cox, L., Dalton, A., Marupadi, V.: Presence-Exchanges: Toward Sustainable Presence-sharing. In: Proc. of 7th IEEE Workshop on Mobile Computing Systems and Applications, Durham, pp. 55–60 (April 2006)

23. Cox, L., Dalton, A., Marupadi, V.: SmokeScreen: Flexible Privacy Controls for Presence-sharing. In: Proc. of 5th Int'l. Conf. on Mobile Systems, Applications and Services, San Juan, pp. 233–245 (June 2007)
24. USDA UV-B Monitoring and Research Program, `http://nadp.nrel.colostate.edu`
25. Eagle, N., Pentland, A.: Eigenbehaviors: Identifying Structure in Routine (2006)
26. Eagle, N., Pentland, A.: Reality Mining: Sensing Complex Social Systems. In: Personal Ubiq. Comp., pp. 255–268 (May 2006)
27. Lamming, M., Bohm, D.: SPECs: Another Approach to Human Context and Activity Sensing Research, Using Tiny Peer-to-Peer Wireless Computers. In: Proc. of 5th Int'l. Conf. on Ubiq. Comp. Seattle (October 2003)
28. Twitter, `http://twitter.com`
29. Burke, J., et al.: Participatory sensing. In: ACM Sensys World Sensor Web Workshop, Boulder (October 2006)
30. Pering, T., et al.: The PSI Board: Realizing a Phone-Centric Body Sensor Network. In: Proc of 4th Int'l. Workshop on Wearable and Implantable Body Sensor Networks, Aachen (March 2006)
31. Sensorplanet, `http://www.sensorplanet.org/`
32. Eisenman, S., et al.: Metrosense project: People-centric Sensing at Scale. In: Proc. of World Sensor Web Workshop, Boulder (October 2006)
33. Reddy, S., et al.: Image Browsing, Processing, and Clustering for Participatory Sensing: Lessons From a DietSense Prototype. In: Proc. of 4th Workshop on Embedded Networked Sensors, Cork (June 2007)
34. UbiFit Project, `http://dub.washington.edu/projects/ubifit`
35. Kansal, A., Goraczko, M., Zhao, F.: Building a Sensor Network of Mobile Phones. In: Proc of IEEE 6th Int'l. IPSN Conf., Cambridge (April 2007)
36. Facebook, `http://www.facebook.com`
37. MySpace, `http://www.myspace.com`
38. Apple iPhone, `http://www.apple.com/iphone/`
39. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
40. Garmin Edge, `http://www.garmin.com/products/edge305`
41. Campbell, A.T., et al.: People-Centric Urban Sensing (Invited Paper). In: Proc. of 2nd ACM/IEEE Annual Int'l. Wireless Internet Conf., Boston (August 2006)
42. Symbian, `http://www.symbian.com`
43. Eisenman, S.B., et al.: The BikeNet Mobile Sensing System for Cyclist Experience Mapping. In: Proc. of 5th ACM Conf. on Embedded Networked Sensor Systems, Sydney, November 6-9, 2007, pp. 6–9. ACM Press, New York (2007)
44. CitySense, `http://www.citysense.net/`
45. Place Lab., `http://www.placelab.org/`
46. Gemmell, J., Bell, G., Lueder, R.: MyLifeBits: a Personal Database for Everything. Communications of the ACM 49(1), 88–95 (2006)
47. Zhou, C., et al.: Discovering Personally Meaningful Places: An Interactive Clustering Approach. In: ACM Trans. on Information Systems vol. 25(3) (2007)
48. Skyhook Wireless, `http://www.skyhookwireless.com/`
49. Liao, L., Fox, D., Kautz, H.: Extracting Places and Activities from GPS Traces using Hierarchical Conditional Random Field. Int'l. Journal of Robotics Research 26(1) (2007)
50. Krumm, J., Horvitz, E.: LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In: Proc. of 1st Int'l. Conf. on Mobile and Ubiq. Systems: Networking and Services (August 2004)

51. Eisenman, S.B., Campbell, A.T.: SkiScape Sensing (poster abstract). In: Proc. of 4th ACM Conf. on Embedded Networked Sensor Systems, Boulder (November 2006)
52. Motorola L2, `http://www.motorola.com/`
53. Lane, N.D., Lu, H., Campbell, A.T.: Ambient Beacon Localization: Using Sensed Characteristics of the Physical World to Localize Mobile Sensors. In: Proc. of 4th Workshop on Embedded Networked Sensors, Cork (June 2007)
54. Sohn, T., et al.: Mobility Detection Using Everyday GSM Traces. In: Proc. of the 6th Int'l. Conf. on Ubiq. Comp. Orange County, pp. 212–224 (September 2006)
55. Anderson, I., Muller, H.: Practical Activity Recognition using GSM Dat. In: Technical Report CSTR-06-016, Dept. of Comp. Sci., Univ. of Bristol (July 2006)
56. iGoogle, `http://www.google.com/ig`
57. Kapadia, A., Henderson, T., Fielding, J.J., Kotz, D.: Virtual Walls: Protecting Digital Privacy in Pervasive Environments. In: Proc. of 5th Int'l. Conf. on Perv. Comp. Toronto, pp. 162–179 (May 2007)
58. Masoodian, M., Lane, N.: MATI: A System for Accessing Travel Itineary Information using Mobile Phones. In: Proc. of 16th British HCI Group Annual Conf., London (September 2002)
59. Google Gadgets, `http://code.google.com/apis/gadgets`
60. Choudhury, T., Basu, S.: Modeling Conversational Dynamics as a Mixed Memory Markov Process. In: Advances in Neural Information Processing Systems 17, pp. 218–288. MIT Press, Cambridge (2005)
61. Hightower, J., et al.: Learning and Recognizing the Places We Go. In: Proc. of the 7th Int'l. Conf. on Ubiq. Comp. Toyko, pp. 159–176 (September 2005)
62. Welbourne, E., Lester, J., LaMarca, A., Borriello, G.: Mobile Context Inference Using Low-Cost Sensors. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, Springer, Heidelberg (2005)
63. Kang, J., Welbourne, W., Stewart, B., Borriello, G.: Extracting Places from Traces of Locations. In: Proc. of ACM Int'l. Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, Philadelphia (October 2004)
64. DeVaul, R., Sung, M., Gips, J., Pentland, A.: MIThril 2003: Applications and Architecture. In: Proc. of 7th Int'l. Symp. on Wearable Computers, White Plains (October 2003)
65. Ashbrook, D., Starner, T.: Using GPS to Learn Significant Locations and Predict Movement across Multiple Users. In: Personal Ubiq. Comp., pp. 275–286 (2003)
66. Liao, L., Fox, D., Kautz, H.: Location-Based Activity Recognition using Relational Markov Networks. In: Proc of IJCAI-05, Edinburgh (August 2005)