Title

CitiSense - Adaptive Services for Community-Driven Behavioral and Environmental Monitoring to Induce Change

Authors

Nikzad, Nima
Ziftci, Celal
Zappi, Piero
et al.

Publication Date

2011-01-18

Peer reviewed

CitiSense – Adaptive Services for Community-Driven Behavioral and Environmental Monitoring to Induce Change

Nima Nikzad, Celal Ziftci, Piero Zappi, Nichole Quick[†], Priti Aghera, Nakul Verma, Barry Demchak, Kevin Patrick[†], Hovav Shacham, Tajana Simunic Rosing, Ingolf Krueger, William Griswold, Sanjoy Dasgupta
Dept. of Computer Science and Engineering, [†]School of Medicine
University of California, San Diego
La Jolla, CA 92093-0404
nnikzad@cs.ucsd.edu

ABSTRACT
In this work we present CitiSense, a new kind of "citizen infrastructure" for the monitoring of pollution and environmental conditions that users are exposed to. By utilizing mobile phones and affordable, small sensors placed in the environment and carried by users, data about pollutants such as ozone and carbon monoxide is collected and used to provide real-time feedback to users and enable them to make healthy changes in their behavior. Results can be reported to a back-end server for further processing and learning, allowing other stakeholders to better understand how diseases such as asthma develop and to help coordinate efforts within a user's community to improve conditions. What differentiates CitiSense from previous projects of this sort is the design of a complete system that addresses issues of mobile power management, data security, privacy, inference with commodity sensors, and integration into a highly extensible and adaptive infrastructure comprising of Open Rich Services (ORS). We discuss the design goals of the CitiSense project, our progress towards the vision of ubiquitous environmental sensing in San Diego, and preliminary results for energy management policies for sensor nodes and mobile phones.

Categories and Subject Descriptors
J.3 [**Life and Medical Sciences**]: Health; C.3 [**Special Purpose and Application-based Systems**]: Real-time and embedded systems; D.2.11 [**Software Engineering**]: Domain-specific architectures, Software architectures; I.5.1 [**Pattern Recognition**]: Statistical Models.

General Terms
Algorithms, measurement, Design, Security, Human Factors.

Keywords
Mobile sensor; air quality; energy management; rich services; service-oriented architecture; real-time feedback.

This research was supported in part by NSF Grant CNS-0932403, as well as a gift from Qualcomm.

1. INTRODUCTION
In our daily activities, we encounter many environmental hazards that are invisible to us, such as pollutants from automobile exhaust, ozone, and methane from landfills and industrial sites. These threats are not isolated to a specific area. According to EPA, 158.5 million people in the US lived in counties that had worse conditions than the national ambient air quality standard in 2007 [22]. In Chula Vista, CA, the incremental cancer risk is 140 cases per million residents, mostly due to diesel exhaust, which is a source of over 40 harmful gasses and cancer-causing substances [23].

The SDAPCD maintains only five air contaminant sampling sites for all of San Diego County, which is 4225 square miles and has 3.1 million residents. The pollutants, such as diesel exhaust, are not uniformly distributed across the county and in time, and the residents of the county are neither equally active at all times nor at equal risk for asthma or other harmful consequences of air pollution.

What if everyday people could be given real-time feedback about the pollutants they were being exposed to during the course of their day? A recent PALMS project study found that overweight patients that were given feedback through their mobile phones about their daily physical activity lost significantly more weight than other patients [1]. Would feedback about pollution exposure lead to people making healthier choices about where they live, work, and play?

In this work we present CitiSense, which is a new kind of "citizen infrastructure" to monitor pollution and environmental conditions that users are exposed to in their daily lives. The system includes mobile phones and affordable, small sensors placed in the environment and carried by users, to collect data about pollutants such as ozone and carbon monoxide. The data is then used to provide real-time feedback to users and allow them to change their behavior for increased health and life quality. The data can be shared with the back-end cyber-infrastructure for further processing, modeling and learning, helping other stakeholders of the system better understand how diseases such as asthma develop and coordinate efforts within a user's community to improve conditions. CitiSense is different from previous projects of this sort in that it includes the design of a complete system that addresses issues of mobile power management, data security, privacy, inference with commodity sensors, and integration into a highly extensible and adaptive infrastructure comprising of Open Rich Services (ORS). PALMS [1] is a similar project that has used such an architecture successfully. CitiSense uses the same core as PALMS and builds on top of it to address more cross-cutting concerns such as power efficiency and immediate user-feedback based on sensed data.

We discuss the design goals of the CitiSense project, our progress towards the vision of ubiquitous environmental sensing through San Diego, and early results for energy management policies for sensor nodes and mobile phones.

The paper is organized as follows. In section two, we discuss related air pollution monitoring projects and in section three we explain why such systems are important to the future of our health. Section four first provides an overview of the complete system and then delves into the details about each sub-system. In section five we conclude with a discussion of future work for the project.

2. RELATED WORK

PALMS-CI [1] is a cyber-infrastructure built at the University of California, San Diego to support the research of a worldwide community of exposure biologists. This community is represented by a number of principal investigators (PIs) that monitor and study human health as a function of geographical location and ambient conditions. Each PI may conduct one or more studies, which typically involves collecting data from sensors (e.g., heart rate, accelerometer, and GPS) worn by scores or hundreds of human subjects for periods of a week or more. Once the data is collected, either the PI or a research assistant (RA) uploads it into a PALMS repository, where it remains available for analysis and visualization. Additionally, the PI may agree to share raw or processed data with other investigators.

The PALMS-CI is a Service Oriented Architecture based on a Rich Services [13] pattern. PALMS-CI services implement the functionality of major domain entities (e.g., calculations, visualization) and services are connected via a message bus. The PALMS-CI presents itself as a single component that exposes its services via Web Services-based API calls using a request/reply pattern.

CitiSense builds on top of the core of PALMS-CI to provide the design and implementation of more cross-cutting concerns such as power management, dynamic computation scheduling, privacy and security. It also provides the infrastructure for real-time filtering, computation and feedback to users.

PEIR [23] is a system that tracks a user locations throughout the day via GPS data collected by their mobile phone and predicts exposure levels based on weather and traffic conditions at those locations. iMAP [24] is a similar system to PEIR, and uses population data in addition to the other sources of pollution data to create exposure predictions. Both solutions are limited in that the only location positional data is gathered from a user and exposure is predicted using previously developed models. In addition, all analysis is done on a back-end server, thus limiting real-time feedback available to a user. In contrast, projects like MobGeoSen [25] gather readings directly from Bluetooth enabled sensors and allow users to tag data with their location and store it until it can later be uploaded to a PC and visualized. Most projects either track a user's location and make predictions or they track pollutants directly, however an opportunity exists to bring together both sources of data to provide more accurate feedback to users as well to create new and more accurate models about pollution in our communities. In addition, energy efficiency is often an afterthought during the development of wireless health systems and often leads to a system with an impractically short battery life.

3. AIR POLLUTION AND HEALTH

Important causes of morbidity and mortality are associated with exposure to both indoor and outdoor air pollution. Outdoor air pollution is estimated to be responsible for 1.4% of total mortality (800,000 deaths), 0.5% of all disability adjusted life years (DALYs), and 2% cardiopulmonary disease [6]. Disability adjusted life years (DALYs) represent the sum of years of potential life lost due to premature mortality and the years of productive life lost due to disability.

World Health Organization (WHO) estimates indicate that the majority of mortality is among the elderly with 81% of attributable deaths and 49% of attributable DALYs occurring in those aged 60 and older [6]. Further, while children under the age of 5 years represent only 3% of total attributable deaths, they are responsible for 12% of total DALYs attributable to poor air quality [6]. It is important to note that these estimates are likely under representative of the true burden of air pollution. Further, in order to fully understand and address the impact of air pollution on human health, it is first necessary to adequately characterize personal exposure for the general population as well as populations at increased risk of adverse outcomes. Populations at increased risk may include the elderly, children, or those with pre-existing medical conditions such as asthma.

Asthma is an inflammatory disorder of the lungs that affects approximately 300 million people worldwide and results in significant individual as well as societal costs [4]. The prevalence of asthma is increasing [4] with concomitant increases in the individual and societal burden associated with the disease. Asthmatic symptoms in specific populations, primarily children, can be exacerbated by exposure to components of air pollution such as ozone and particulate matter [7]. Thus, in order to reduce the burden of suffering from asthma, exacerbating factors such as air pollution must be considered. Research in this area includes studies addressing the relationships between air pollution and the incidence of new cases of asthma as well as the severity of existing asthma. This research will benefit from improved methods of exposure modeling that can more accurately characterize personal exposure in time and by location.

Health Effects

The US Environmental Protection Agency (EPA) has set National Ambient Air Quality Standards (NAAQS) for various components of air pollution with demonstrable detrimental effects on human health and the environment. The following are the six "criteria" pollutants for which there are NAAQS: carbon monoxide, ozone, particulate matter, nitrogen dioxide, sulfur dioxide, and lead [2]. There are numerous health effects of these pollutants including increased morbidity and mortality.

Exposure to ozone and particulate matter has been shown to worsen asthma symptoms [7]. Further associations between particulate matter exposure and negative health outcomes include: increased mortality, increased urgent care visits, and increased hospital admissions for cardiovascular and respiratory disease [6]. Similarly, epidemiologic studies demonstrate associations between exposure to sulfur dioxide, nitrogen dioxide, and carbon monoxide and increased cardiopulmonary mortality and hospital admissions [5]. Despite evidence regarding the negative health effects of air pollution, more research is necessary to fully understand the impact of air pollution on health and disease outcomes. Further, improved understanding of individual exposures to air pollutants can help characterize risk for adverse events in specific populations such as the elderly and those with pre-existing medical conditions.

Real-time Assessment and Feedback on Air Quality

Data from wearable, mobile or environmentally embedded wireless devices yield the potential to understand and influence human behavior within the context of real world social and environmental interactions. Most research involving these devices to date has focused on ecological momentary assessment (EMA) that leverages the ability of wireless devices to capture data points with little interference in the daily life of the person [19]. While EMA systems that can gather, process, and analyze multiple streams of data will be in high demand as sensing technology improves, there is a concurrent demand emerging for systems that also support ecological momentary intervention (EMI) – the use of one or more types of data collected in real-time from free-living individuals to elicit a desired health outcome, such as a prompt for a health behavior or to use a health device (e.g. preventive use of an asthma inhaler in the presence of toxic air pollutants) [20]. The CitiSense system aims to cater to these demands.

4. SYSTEM OVERVIEW

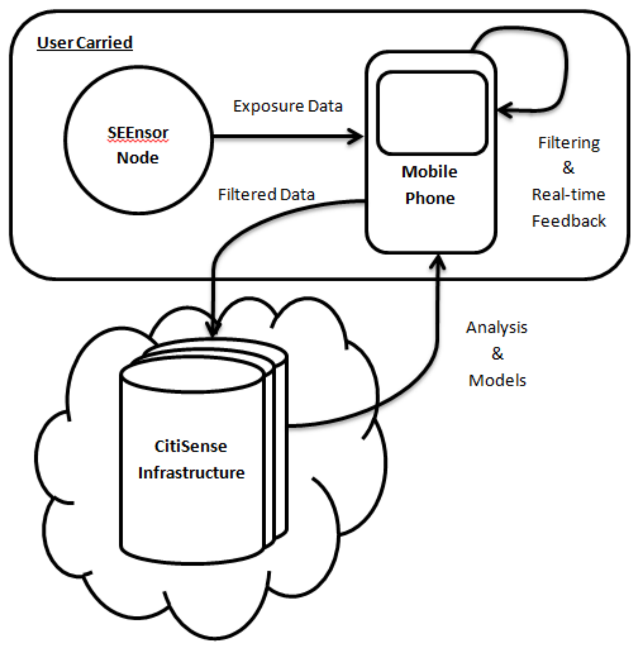

Figure 1. A simple overview of the CitiSense system.

CitiSense encompasses the collection, filtering, analysis, and presentation of air quality data while providing real-time feedback and suggestions to users. For the collection of pollution exposure data, we have built a sensor equipped embedded platform, designed to be worn by users, that we call the *SEEnsor node*. This data is reported to a user's mobile phone through a Bluetooth connection, where it is filtered and briefly analyzed to provide *real-time feedback*, warnings, and suggestions to users to help limit their exposure to pollutants. The amount of processing that should happen at the sensor node, mobile phone, and back-

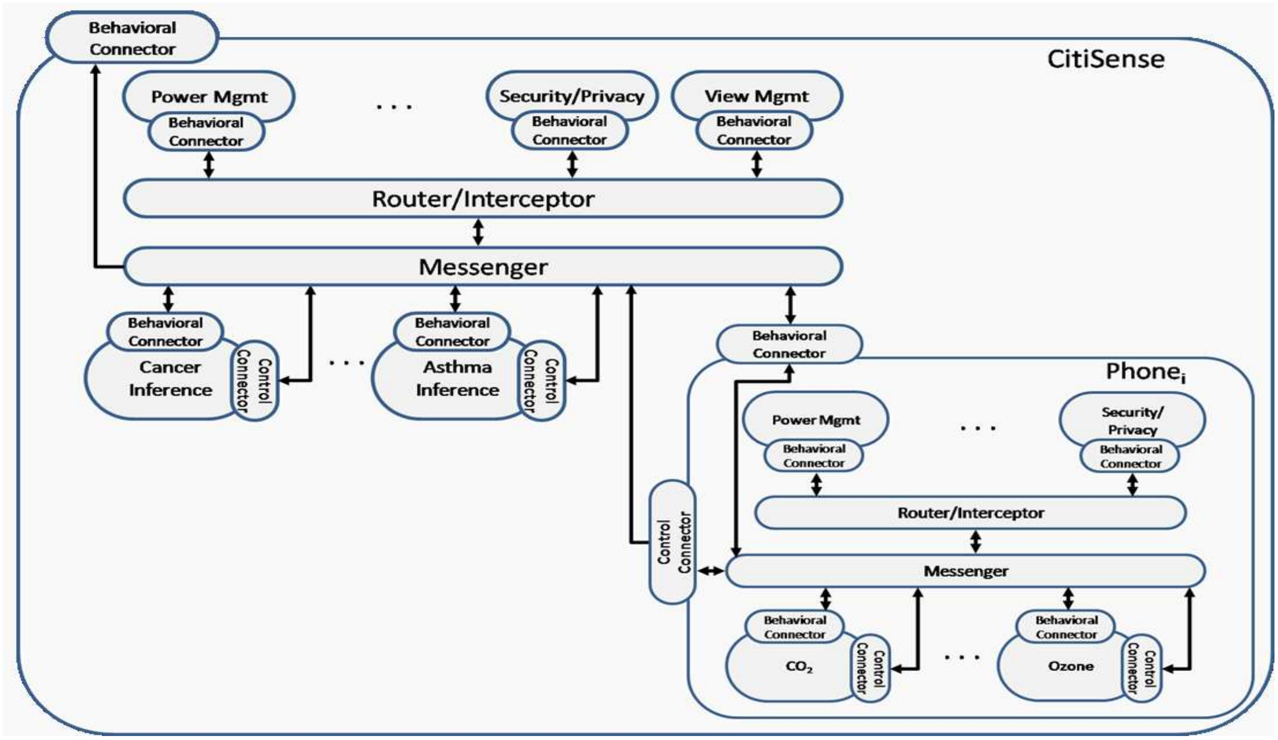

Figure 2. Rich Services architectural pattern as applied to CitiSense.

end server is dynamically adjusted to maximize system battery life through the use of *energy efficient task allocation*. Users can report data and request additional analysis and predictive models through the use of services provided by the back-end infrastructure.

In this section, we present details about each sub-system that makes up the CitiSense project, starting with a discussion of the cyber infrastructure and service oriented architecture, followed by details about our wearable embedded sensor node, filtering and real-time feedback provided by mobile phones, and algorithms for energy efficient task assignment.

4.1 Cyber-Infrastructure

A cyber-infrastructure (CI) is an Internet-based collection of computing services dedicated to providing data storage, computations, and visualizations to a stakeholder ecosystem. A major CI function is to execute workflows on behalf of stakeholders.

As an emerging class of large scale computing systems, cyber-infrastructures (CIs) are poised to become important enablers of community-based computational and information processing in academia, government, and commerce. As an Internet-based distributed collection of data storage, computation, and visualization resources, CIs provide a substrate on which stakeholder communities can build and deliver value by organizing CI resource access through automated processes called workflows. They also provide an infrastructure through which communities can create significant additional value via cooperation and exchange.

An important approach to realize CIs is Service-Oriented Architectures (SOA), which, at its core, represents computing activities as patterns of interaction between computing components where information is exchanged via messages. By specifying interactions between components representing CI resources, a SOA can be used to model CI workflows. A critical feature of SOA systems is the ability to intercept messages traveling between components – thus enabling message transformations and additional message routing that can respond to stakeholder requirements by altering or augmenting workflows without compromising existing functionality.

Service Oriented Architecture

Our approach to the design of CitiSense includes the analysis of the producers of data, the consumers of data, and the operations services on that data in a Service Oriented Architecture (SOA).

The term *service* is often used for Web Services, and the term *Service Oriented Architecture* for the use of Web Services to create applications that are based on the Internet and leverage existing standards-based interaction technologies (such as HTTP/SOAP, XML, WSDL etc) between different entities. These technologies enable the construction of distributed and loosely coupled systems. However, they do not address the broader problems of understanding the relationships between those entities, addressing cross-cutting concerns that relate to all of those entities on different levels, and designing systems that leverage the relationships and the cross-cutting concerns on the entities at the same time.

In CitiSense, we use these terms on a more basic and generic level. A service is a choreography of the interactions between entities [14]. Services are independent of any particular implementation technology [15]. The description of a service is focused on the identification of the roles played by the entities, the interactions between those roles, and the cross-cutting properties important for the system. A Service Oriented Architecture is one that models the roles and their interactions. Creating a SOA involves identifying roles and the services that they are involved. A SOA can then be a model of either the logical level, or the deployment level or both of the system.

On the logical level, a SOA models roles and their interactions independent of the technology that will be used to realize them. It decomposes the system into services that are well-defined and encapsulated. These services provide the immediate and long term business goals of the system. Each of these services can be composed of other services, can act as a proxy for other services, or can be a stub for another service. Service oriented analysis provides identification of services as well as cross-cutting concerns of the system, which provides reliability, extensibility and maintainability for the system in the long run. An example of a cross-cutting concern in CitiSense is power efficiency, which exists on sensor, cell-phone and cloud levels, with different characteristics, but with a common goal of optimizing the power efficiency of the overall system.

On the deployment level, a SOA models services as interactions of self-contained, loosely coupled, standards based components to implement the roles modeled earlier on the logical level. Such components, similar to the logical level, can be composed of other components, can act as proxies for other components or can be a stub for another component. In CitiSense, the interaction of components occurs via messaging.

At both the logical and the deployment levels, SOAs encourage manageability, maintainability, scalability, interoperability, composition, incremental development and testability.

Open Rich Services

We based the design of CitiSense on Open Rich Services architectural pattern and development model. This architectural pattern leverages the composite pattern [12] and the messaging and routing patterns [13] to create a system of systems. The development model of Open Rich Services provides the means to identify roles based on application requirements, and services that involve the interactions of those roles, thus leading to both logical and deployment level SOAs. It also provides an effective process to identify cross-cutting concerns early in the development, and incorporate necessary components for those concerns early in the development.

A Rich Services architecture organizes systems-of-systems into a hierarchically decomposed structure that supports "horizontal" and "vertical" services integration. The horizontal integration involves managing the interaction of application services and cross-cutting concerns on the same level. On the other hand, vertical integration involves the hierarchical decomposition of an application service into a set of sub-services. The environment of the sub-services, their composition and interaction, and their structural/behavioral complexity are covered inside the application service, and hidden from the other application services on the same level (see figure 2).

There are three main entities in a Rich Services architecture: Rich Services, Messenger and Router/Interceptor, and Behavioral/Control Connectors. A Rich Service provides the application and infrastructure functionality. A Rich Service can be composed of other Rich Services. The Messenger and the Router/Interceptor provide the infrastructure for the communication of Rich Services at the same level internally. The Behavioral/Control Connector serves as the means for a Rich Service to communicate with its environment.

A Rich Service can be a simple functionality block such as a Commercial Off The Shelf (COTS) system or a Web Service, or it could be hierarchically decomposed of other Rich Services. We distinguish between *Rich Application Services* (RAS) and *Rich Infrastructure Services* (RIS). RASs are directly connected to the Messenger, and they provide core application functionality, such as data storage and environmental sensing. RISs are directly connected to the Router/Interceptor, and provide common infrastructure and cross-cutting concerns functionality, such as power efficiency management and authorization.

The Messenger and Router/Interceptor provide horizontal integration. They route messages between Rich Services on the same level (RASs and RISs).

The *Messenger* layer supports decoupling between services by providing a messaging medium for services to communicate.

The *Router/Interceptor* leverages the interceptor pattern [16], and intercepts messages carried by the Messenger and reroutes them appropriately. This provides the facility to inject dynamic behavior onto the interactions between Rich Services on the same horizontal level, such as the cross-cutting behaviors.

The Behavioral/Control Connectors are the means by which a Rich Service connects to and interacts with its environment. They encapsulate and hide the internals of the

Rich Service, and allow the exporting of only the interface that the Rich Service would like to expose and provide. They allow identifying the interfaces provided by each Rich Service, and help to simplify the systems-of-systems integration challenge.

The *Behavioral Connector* allows the Rich Service to expose what service it provides, thus opening the way for data manipulations.

The *Control Connector* allows the Rich Service to expose how it provides its service, thus providing the means for changing its settings.

In this architecture, each Rich Service can be a simple block or decomposed down into other Rich Services, while the Behavioral/Control Connector is the only mechanism that makes routing messages between different vertical levels possible. In addition to this, the use of the Messenger and Router/Interceptor layers removes the hard dependency of services on each other, while abstracting away the relative locations of each service in the logical hierarchy. This way, services from different hierarchy levels can interact with each other seamlessly, without being aware of each other's locations and dependencies.

4.2 Air Quality Sensing

4.2.1 The SEEnsor Node

The System Energy Efficiency sensor Node (SEEnsor Node) is a flexible platform developed for the rapid prototyping of wireless sensor network applications. This system has a modular architecture made up of a set of 2x2 inches layers that can be independently replaced in order to quickly test and compare different solutions for a specific application. The current platform is a prototype and will eventually be shrunk down to a smaller size. By combining different layers, developers can easily explore the design space to find the optimal trade-off between a variety of metrics, such as energy consumption, computational power, communication throughput, and communication latency. Figure 3 presents the node architecture.

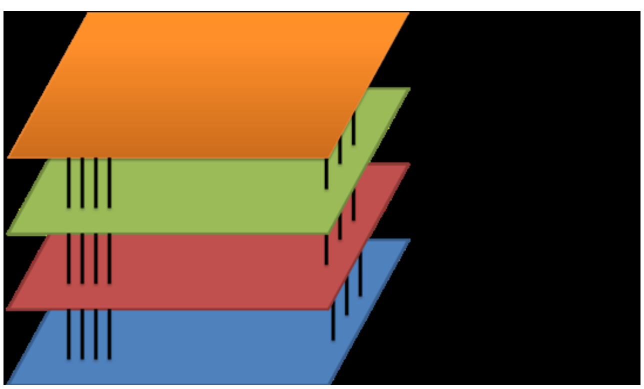

Figure 3. The SEEnsor node architecture.

At the current stage of development, the communication layer includes a Bluetooth transceiver, the microcontroller layer an 8-bit microcontroller and the sensor layer all the necessary electronics to interface with a carbon monoxide and ozone sensor board.

Figure 4. The current SEEnsor prototype.

4.2.1.1 Communication Layer

The communication layer is responsible for the forwarding of collected data from the sensor node to an external system. Our current design uses the WT12 Bluetooth transceiver from Bluegiga [9]. This module is a complete Bluetooth solution and includes a system on chip (microcontroller and transceiver on the same die), a ceramic antenna and all the necessary circuitry to interface them. Bluetooth was selected as the wireless technology as it pairs with most modern mobile phones and is approved for patient trials. The microcontroller is loading the iWRAP firmware that is used to easily interface the module to another host microcontroller through a set of string based commands issued through the UART interface. This masks all the complexity of the Bluetooth protocol from the developer.

While Bluetooth is a power hungry protocol, the module can operate in low power states when it is not connected to a piconet, significantly reducing the average power consumption of the system. Table 1 presents the power consumption characteristics of the WT12 transceiver.

Table 1. Power consumption characteristics of the Bluetooth radio.

State	Power consumption
Transmit/Receive (avg.)	30 mA
Transmit/Receive (max)	70 mA
Transmit (low power)	20 mA
Idle (no connection)	1.2 mA

4.2.1.2 Microcontroller Layer

The microcontroller layer is the core of the SEEnsor node. The current implementation includes the 8-bit ATMEGA1281 microcontroller from ATMEL and a 7.3728MHz crystal [8]. This low power, yet powerful, microcontroller features 128kB of built-in Flash for both code and data, 8kB of RAM and 4kB of EEPROM. This MCU features a RISC architecture that can work at up to 16MHz while achieving low power consumption. For example, at 8MHz the power consumption at 3.3V is about 7mA. The ATMEGA1281 can operate in several reduced power states to save energy when possible. These sleep modes present current consumption of several degrees of magnitude lower than in the while in the active state. Table 2 presents the typical current consumption rates for the ATMEGA1281.

Table 2. Typical current consumption in different sleep modalities and speeds when the MCU is powered at 3.3V.

Frequency	Active mode	Idle mode	Sleep mode
1 MHz	1 mA	0.25 mA	6 uA
4 MHz	4 mA	0.8 mA	6 uA
8 MHz	7 mA	1.5 mA	6 uA

4.2.1.3 Power Layer

The power layer is responsible for regulating the input source of energy and to power the other layers of the node. At the current stage of development, the sensor node is powered by a 3.7V, 1800mAH Li-ion battery or a USB cable.

This layer includes the LTC3553 battery manager from Linear Technology [10]. The task of this low power (12 uA stand by quiescent current), low cost device is two-fold. It not only regulates the power coming either from a USB connection or the battery to produce two distinct voltage levels, but it also manages the battery charge.

The LTC3553 includes a buck regulator and a LDO regulator, each providing a 3.3V output. Due to its higher efficiency and maximum output current (200 mA), the buck regulator output is meant to be used to power most of the circuit. On the other hand, due to the lower level of output noise, the LDO is meant to power modules that require higher noise rejection on the power line, such as a wireless transceiver.

In addition to the regulated outputs, one pin in the connector is reserved to forward 3.7V from the battery in case different voltages are needed by the other layers. This line can be used instead of a regulated one to feed additional voltage regulators to improve system energy efficiency.

The control signals that enable the outputs of the LTC3553 are also available to the microcontroller to reduce power consumption when parts of the node can be powered down.

4.2.1.4 Sensor Layer

Currently, the sensor layer includes all the circuitry needed to interface evaluation boards for two toxic gases: Carbon Monoxide (CO) and Ozone (O_3).

For our initial studies, we plan to monitor low level pollutants typical of open air environment that should (hopefully) be lower than the safe exposure limits defined by the Environmental Protection Agency [2]. Our system uses the 3E/F electrochemical carbon monoxide sensor from City Technologies [11] and is able to detect CO concentrations as low as 0.5 ppm (parts per million). The O3 3E1 electrochemical ozone sensor, also from City Technologies, is able to sense O_3 concentrations as low as 20 ppb (parts per billion). By comparison, the EPA air quality standard for carbon monoxide is 9 ppm, while the standard for ozone exposure is 75 ppb.

4.2.2 Real-time Feedback

A user's mobile phone is paired to a carried SEEnsor node and is streamed sensor readings for the pollutants of interest. The data is first filtered to remove any possibly faulty or useless values, and is then aggregated. The data is quickly analyzed as it arrives to identify any patterns or readings of concern, such as a prolonged exposure to high levels of carbon monoxide. The patterns and warning signs that are watched for may be specific to a patient with a certain condition (an asthmatic will be more sensitive to ozone and should therefore be alerted at lower levels of exposure) or common to all participants. Detecting unsafe conditions must happen in real-time to be helpful, therefore some level of analysis and filtering of the data must happen at the mobile phone. In most cases, users will be exposed to pollutants at levels that are deemed safe. This data is still useful for learning models and identifying patterns, so it is beneficial to share it and compare it to readings of other users in the region. Data is uploaded from the mobile phone to the cyber-infrastructure and stored, tracking where the reading is from (using GPS data) and at what time the reading was taken. By first aggregating the data and providing some filtering and analysis before transmitting the data to the back-end, the amount of data to be transmitted can be significantly reduced and drastically improve battery life. After the data has been analyzed and shared with the back-end, it can be cleared from the user's phone. A user is able to fetch previous exposure data in the future from the cyber-infrastructure through its provided services.

A significant contribution of our project will be its focus on providing real-time feedback to users about the amount of pollution they are exposed to and providing suggestions to promote healthy behavioral change. To do so, finding new ways of presenting and visualizing pollution data will be very important. An application targeting users that run outdoors often will use collected samples and computed models to determine the running path from point A to point B that minimizes the amount of exposure to air pollution. If the user is carrying a pollution sensor, the path can be updated and feedback can be given to a user in real-time, suggesting changes to how and where a user runs based on sensed pollutants. For young patients, exploring the use mobile phone games may present new ways of making health data interesting to children. The collected pollution data will only be as useful to users if it is presented in a clear and interesting fashion.

4.3 Power Management

Mobile healthcare systems have a heterogeneous and tiered architecture consisting of a set of wireless sensors, single wireless local aggregator (often a mobile phone) for each user and a backend server. The main mode of operation is to sense levels of various pollutants in the environment and gather it on a mobile phone, where the aggregated data can be sent to the back end or decide to process it locally depending upon its processing power. All of these components use different wireless radios for communication and have varying processing power. Sensors and mobile phones are battery powered, while the back-end has unlimited power. Conserving battery energy on sensors and cellphone in such wireless healthcare system is a big challenge. Thus, in this project we also try to address this challenge of energy management by designing dynamic algorithms to maximize system battery life.

We plan to leverage the increased processing ability of today's sensors and cellphones to reduce the energy cost of wireless transmission. Thus, it would address the tradeoff between local processing and communication by deciding how much processing to do on which component of the system. This decision depends on various factors like the processing capability of the device, the wireless connectivity between devices and, above all, the battery charge of the devices. In other words, the goal of our algorithm is to determine an energy efficient task assignment that would maximize the system battery life. We define system life as the time from the start of the system until the time the first battery operated device dies. We will use a static integer linear program solution as a baseline for comparison with our dynamic solution. The ILP solution provides optimal results for a known set of tasks assuming the workload and system characteristics do not change at run time. In the static case, optimal assignment of tasks to heterogeneous nodes in the system significantly affects the performance and battery life of the system. The system characteristics like wireless channel characteristics typically change at runtime, differing system response to the sensed data, the depletion of the battery of some components may change the dynamics. Thus, we need a dynamic algorithm that adapts to such changes at runtime.

Efficient Task Assignment

We model the mobile healthcare application as a directed acyclic task graph (DAG): G = (T, E) in which each vertex represents a task $T_i \in T$. Each edge in the graph $E_{ij} \in E$ shows that task T_j is dependent on T_i in order to perform computation. The weights W_{ij} on edge E_{ij} represent the amount of data to be communicated from task T_i to T_j. In order to maximize system life our goal is to map these tasks onto the three tier architecture such that the system would last longest. As a baseline for comparison we will formulate an integer linear program (ILP) similar to one done by P. Aghera et.al in [21]. In simulation results of the DynAHeal algorithm running on a mobile phone with dynamic workloads, battery life was improved by up to ~35% in comparison to the design time task assignment given by the ILP.

4.4 Statistical Modeling of Pollution Data

For studying the effects of pollution in a given community, it is necessary to have an accurate pollution map of the area. The principal goal of producing this map is to understand how pollutant levels vary across different regions, and how the pollution map as a whole evolves over time. Consider, for instance, a private citizen Jane who has a history of asthma planning to take a trip visiting different parts of the city during the day. Having a system capable of predicting potentially hazardous pollutant levels could help Jane, plotting out a route that minimizes her exposure to the pollutants. She may even consider visiting certain parts of the city at specific times when she knows that the pollution level is relatively less. Due to these specific needs, we require a model that can not only be able to interpolate the measured pollutant values from noisy sensors (both stationary as well as mobile) but also has the capability for predicting the value at some future time instance. We thus propose to model the system as a spatio-temporal process.

4.4.1 Pollution Modeling Using a Spatio-Temporal Process

Given a specific region of interest, the basic idea is to model the pollutant level in the region as a Gaussian Process using a technique called Gaussian Process Regression (GPR) [3]. This gives us the ability to effectively interpolate the data measured from various sensors at a particular time instance. GPR alone, however, falls short in accurately predicting pollutant levels at some future instance in time. We thus augment our GPR model by adding an auto-regressive (AR) time-series process that can effectively model the daily and seasonal cycles in pollutant levels at a particular location. Since pollutants levels in different locations can evolve differently, we add different AR processes at different locations.

Learning the Model Parameters

The sensed pollution data is first interpolated spatially using the GPR model. Once we have an estimate for each location, it is used in training the temporal AR processes (each temporal process corresponding to a different location).

Predicting Pollutant Levels from Learnt Model

Given a specific query location at some future time, we extract the AR process corresponding to that location and make the prediction according to the pollutant level predicted by that process at the future time.

4.5 Security Issues

The data streams generated by individual users have serious privacy implications, even if explicit location info is removed. Consider that some places may have unusual or even unique combinations of readings for various pollutants such as houses with an unusual mold or fields near a factory. In such cases, it may be possible to infer a user's location from the 'sensor signature' of his location. If an attacker simply wishes to confirm that his target was in a particular (known) place, his task might be even easier. Or he might intentionally release particulates into the air at a certain place to create a pollutant signature for it; every user who passes by can now be identified.

One way to balance the privacy needs of individuals and the data analysis needs of public health officials is to collect readings into a database and then aggregate and perturb the readings; the hope is that enough information can be obscured that individual privacy is not compromised, but not so much that the data analysis is inaccurate. Much recent work in theoretical cryptography has focused on the possibility for such privacy-preserving data mining [17].

At the same time, practical data anonymization techniques are often found lacking. In two recent examples, researchers were able to identify individuals in supposedly anonymized search logs released by AOL and a movie-rating database released by Netflix [18].

4.6 Overview of System Operation

To make things clearer on how using the Open Rich Services architecture provides application services and cross-cutting concerns interact on different levels, we can consider how privacy and power are managed within the system across the hierarchical levels in figure 2. The figure shows the backend infrastructure labeled as "CitiSense", and one of the many cell phones connected to it with "Phone_i". The phone is the component that contains the interaction of sensors with the environment. The environment is sensed by the sensors at the bottom in figure 1. The sensor data is put onto the Messenger wrapped inside a message to be stored by another service that handles storage tasks. The message on the Messenger component is intercepted by the Router/Interceptor component, which passes it onto the Security/Privacy and Power Management services. The Security/Privacy service first decides whether to allow the storage of the sensed data, which is a decision depending on the preferences of the user carrying the sensor. If the user chooses not to do that, the data is either dropped or stored locally for a limited amount of time for local processing (eg. to provide feedback to the user). Assuming the user allows this data to be shared with the backend infrastructure, the Power Management service gets its turn to decide on where the data should be stored: locally or in the backend infrastructure, based on the current remaining battery life on the cell phone. The data can either be sent to the backend server immediately, consuming more power, if the remaining battery is more than a certain threshold; or data can be buffered until it reaches a certain total amount, after which it is sent to the backend infrastructure in a batch so as to consume less battery. These are all decisions that are made on the phone level. If the data is sent to the CitiSense backend infrastructure, similar decisions need to be made on that level as well. The data might be stored after anonymization due to privacy concerns, or it might be stored in a nearby data storage facility initially and moved to a common storage space afterwards to use less power in the backend infrastructure. Although these cross-cutting concerns exist with different parameters on different levels, the Open Rich Services architecture provides a hierarchical, extensible, scalable, manageable framework to handle the challenges that come with the CitiSense system.

5. CONCLUSION

Going forward, the SEEnsor node will eventually be modified and adapted to long term deployments in the environment as a stationary sensor node, with larger and more sensitive sensors, long range radio, and solar panels to harvest energy. As the number and types of sensors grows with the project, we will be able to develop new energy management policies for our sensors and new predictive models for pollution levels. As we collect more data, we will be able to validate our models and ensure that they scale well to large data sets. Online learning techniques will be critical to dynamically update our models as the system grows.

In this paper, we presented our vision for and progress towards a "citizen infrastructure" for monitoring air pollution. In the coming months, a complete end-to-end prototype will be completed and tested in the field. This prototype will include wearable SEEnsor nodes, a mobile phone with applications for reporting sensed values and giving feedback and suggestions to users, and the cyber-infrastructure to store and process the incoming data. We are currently planning a study where users that cross daily at the U.S.-Mexico border near San Diego will carry a SEEnsor node and not only collect valuable data about the levels of pollution that commuters are exposed to, but will also provide valuable feedback regarding how to best present this information to users to make the system informative and helpful.

6. REFERENCES

[1] UCSD PALMS Project (2010). http://ucsd-palms-project.wikispaces.com/.

[2] U.S. Environmental Protection Agency: National Ambient Air Quality Standards (2010). http://www.epa.gov/air/criteria.html

[3] Rasmussen, C. E. and Williams, K. I. 2006. Gaussian Processes for Machine Learning. MIT Press.

[4] Bahadori, K., et al. Economic burden of asthma: a systematic review. BMC Pulmonary Medicine 2009, 9:24.

[5] Bernstein, J. Health effects of air pollution. J Allergy Clin Immunol 2004;114:1116-23.

[6] Ostro B. Outdoor air pollution: Assessing the environmental burden of disease at national and local levels. Geneva, World Health Organization, 2004 (WHO Environmental Burden of Disease Series, No. 5).

[7] Romieu I, Meneses F, Ruiz S, et al. Effects of air pollution on the respiratory health of asthmatic children living in Mexico City. Am J Respir Crit Care Med 1996; 154: 300 ± 307.

[8] Atmel Corporation (2010). http://atmel.com/.

[9] Bluegiga Technologies (2010). http://www.bluegiga.com/.

[10] Linear Technologies (2010). http://www.linear.com/.

[11] City Technology (2010). http://www.citytech.com/.

[12] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

[13] Hohpe, G. and Woolf, B. 2004. Enterprise Integration Patterns: Designing, Building, and Deploying. Addison-Wesley Professional.

[14] Krüger, I. H. 2004. Service Specification with MSCs and Roles. Proceedings of the IASTED International Conference on Software Engineering (IASTED SE'04), Innsbruck, Austria.

[15] Krüger, I. H., Mathew, R., Meisinger, M. 2006. Efficient Exploration of Service-Oriented Architectures Using Aspects. *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*, Shanghai, China.

[16] Schmidt, D., Stal, M., Rohnert, H., and Buschmann, F. 2000. Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects. Wiley & Sons.

[17] Dwork, C. 2008. Differential Privacy: A Survey of Results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation (TAMC)*, Apr. 2008. Invited Paper.

[18] Narayanan, A. and Shmatikov, V. 2008. Robust De-anonymization of Large Sparse Datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (May 18 - 21, 2008). SP. IEEE Computer Society, Washington, DC, 111-125.

[19] Stone, A.A., Shiffman, S. Capturing momentary, self report data: a proposal for reporting guidelines. *Annals of Behavioral Medicine, 24,* 3 (2002), 236-243.

[20] Patrick, K., Intille, S., Zabinski, M. An ecological framework for cancer communication: Implications for research. *J of Medical Internet Research, 7,* 3 (2005), e2.

[21] Aghera, P., Coskun, A., Fang, D., Krishnaswamy, D., Simunic Rosing, T. DynAHeal: Dynamic energy efficient task assignment for wireless healthcare systems. DATE10.

[22] Environmental Protection Agency, "National Air Quality – Status and Trends through 2007", Highlights. http://www.epa.gov/air/airtrends/2008/report/Highlights.pd

[23] San Diego Air Pollution Control District, "Air Quality in San Diego – 2007 Annual Report". http://www.sdapcd.org/info/reports/ANNUAL.pdf

[24] Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., and Boda, P. 2009. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international Conference on Mobile Systems, Applications, and Services* (Kraków, Poland, June 22 - 25, 2009).

[25] Demirbas, M., Rudra, C., Rudra, A., and Bayir, M. A. 2009. iMAP: Indirect measurement of air pollution with cellphones. In*Proceedings of the 2009 IEEE international Conference on Pervasive Computing and Communications* (March 09 - 13, 2009). PERCOM. IEEE Computer Society, Washington, DC, 1-6.

[26] Kanjo, E., Benford, S., Paxton, M., Chamberlain, A., Fraser, D. S., Woodgate, D., Crellin, D., and Woolard, A. 2008. MobGeoSen: facilitating personal geosensor data collection and visualization using mobile phones. *Personal Ubiquitous Comput.* 12, 8 (Nov. 2008), 599-607.