12-2011

Design and implementation of an open framework for ubiquitous carbon footprint calculator applications

Farzana Rahman
Marquette University, farzana.rahman@marquette.edu

Casey O'Brien
Marquette University, casey.obrien@marquette.edu

Sheikh Iqbal Ahamed
Marquette University, sheikh.ahamed@marquette.edu

He Zhang
Tsinghua University

Lin Liu
Tsinghua University

Recommended Citation

Rahman, Farzana; O'Brien, Casey; Ahamed, Sheikh Iqbal; Zhang, He; and Liu, Lin, "Design and implementation of an open framework for ubiquitous carbon footprint calculator applications" (2011). *Mathematics, Statistics and Computer Science Faculty Research and Publications*. 43.
https://epublications.marquette.edu/mscs_fac/43

Sustainable Computing: Informatics and Systems, Vol. 1, No. 4 (December 2011): 257-274. DOI.

Design and Implementation of An Open Framework for Ubiquitous Carbon Footprint Calculator Applications

Farzana Rahman
Marquette University, Milwaukee, WI
Casey O'Brien
Marquette University, Milwaukee, WI
Sheikh I. Ahamed
Marquette University, Milwaukee, WI
He Zhang
School of Software, Tsinghua University, Beijing, China
Lin Liu
School of Software, Tsinghua University, Beijing, China

Abstract

As climate change is becoming an important global issue, more and more people are beginning to pay attention to reducing greenhouse gas emissions. To measure personal or household carbon dioxide emission, there are already plenty of carbon footprint calculators available on the web. Most of these calculators use quantitative models to estimate carbon emission caused by a user's activities. Although these calculators can promote public awareness regarding carbon emission due to an individual's behavior, there are concerns about the consistency and transparency of these existing CO_2 calculators. Apart from a small group of smart phone based carbon footprint calculator applications, most of the existing CO_2 calculators require users to input data manually. This not only provides a poor user experience but also makes the calculation less accurate. The use of a standard framework for various carbon footprint application developments can increase the accuracy of overall calculations, which in turn may increase energy awareness at the individual human level. We aim for developing a carbon footprint calculation framework that can serve as a platform for various carbon footprint calculator applications. Therefore, in this paper, we propose a platform-agnostic *Open Carbon Footprint Framework* (*OCFF*) that will provide the necessary interfaces for software developers to incorporate the latest scientific knowledge regarding climate change into their applications. OCFF will maintain a clouded knowledge base that will give developers access to a dynamic source of computational information that can be brought to bear on real-time sensor data. Based on the OCFF platform, we developed a *Ubiquitous Carbon Footprint Calculator application* (*UCFC*) that allows the user to be aware of their personal carbon footprint based on their ubiquitous activity and act accordingly. The major contribution of this paper is the presentation of the quantitative model of the platform along with the entire design and implementation of UCFC application. We also present the results, analysis, and findings of an extensive survey that has been conducted to find users' awareness of increased carbon footprint, feature requirements, and expectations and desires to alleviate CO_2 emissions by using a footprint calculator. The design of UCFC application incorporates the analysis and inferences of the survey results. We are also developing a fuel efficient mobile GPS application for iPhone suggesting the greenest/most fuel efficient route to the user. In this paper, we also point out some important features of such an application.

Keywords

CO_2 emission, Context, Carbon footprint, Smart software, Open development platform, Ubiquitous, Sensor

1. Introduction

Designing smart environments and software [9], [10], [18] is a goal that appeals to researchers in a variety of disciplines including artificial intelligence, pervasive and mobile computing, middleware, sensor networks, etc. Advances in these supporting fields have prompted a tremendous increase in the amount of smart software for different practical circumstances such as: smart home, weather forecasting, elderly care, personalization, smart interaction, and energy control. One area that has drawn the attention of smart software researchers recently is climate control or energy control. Consequently, global warming and the reduction of carbon dioxide emissions are at the top of the environmental policy agenda today [21], [26].

To address this issue, smart software needs to be designed that can calculate the energy consumption of a user, business organization, and enterprise as a whole. Because of increasing concern about carbon emissions, many companies and organizations are pursuing *carbon footprint* projects to estimate their own contributions towards global climate change. At the same time, with growing awareness of elevated CO_2 levels and climate change, attention is turning to individual behavior as a source of global carbon emission. Every person contributes CO_2 into the air in one way or another. For example: respiration, public transportation, electricity, heating, cooling, recreational activities, etc.—all contribute to increased CO_2. According to Shui et al. [22], consumers in aggregate were responsible for 28% of US energy consumption and 41% of US CO_2 emissions in 1997.

In response to the focus on individuals, numerous websites have been created to help calculate an individual's carbon footprint or provide an estimate of the CO_2 emissions that an individual is directly responsible for over a given period of time. These calculators typically divide the individual's profile into household activities and transportation. Based on differing formulations of user input they produce a quantified amount of CO_2 per year. These calculators are provided by government agencies, non-governmental organizations, and private companies. Google PowerMeter [15] receives information from utility smart meters and energy management devices and provides customers with access to their home electricity consumption right on their personal iGoogle homepage. The Carbon Diem [6] project utilizes GPS and mobile phones to give accurate usage of CO_2 for fixed travel patterns. However, the recent rise in carbon calculators has been accompanied by inconsistencies in output values given similar inputs for individual behavior. A comparative study is analyzed in [18].

Despite of all these facts, people are interested in making personal changes to reduce their contribution to climate change. We focus our efforts on these people who are actively seeking to reduce their carbon footprint. For such users, tracking energy usage for each dynamic activity (electricity usage, driving in long route, etc.) is as important as getting suggestions to reduce carbon footprint by rational decision making. This approach allows users to prioritize among the many possible ways they can reduce their environmental impact. These features are not addressed by any of the calculators developed so far. Therefore, in this paper, we present the design and implementation of an Open Carbon Footprint Framework (OCFF) that includes the above mentioned two important features.

1.1. Our major contributions

- We formally define “carbon footprint” from a ubiquitous computing perspective and analyze its properties.
- We propose an *Open Carbon Footprint Calculation Framework* (*OCFF*) that collects various context data from sensors and provides the necessary interfaces for software developers to incorporate the latest scientific knowledge regarding climate change into their applications.
- OCFF holds a set of simple calculation models that is maintained by experts.
- In order to demonstrate the use of OCFF, we developed a *Ubiquitous Carbon Footprint Calculator application* (*UCFC*) that determines *ubiquitous carbon footprint* based on users’ *dynamic activity*.
- We present the design challenges of developing a UCFC application in this paper. UCFC is different from the already existing calculators as it uses the built-in sensors of mobile devices to determine user activity.
- Using the collected sensor data, UCFC calculates an approximate carbon footprint by summing up the results of the applicable models. The implementation details and usability study of UCFC are also presented in this paper.
- We present the details of a computational model for UCFC application that can estimate CO_2 consumption due to a person's dynamic activity within a room's environment.
- We also present the results, analysis, and findings of an extensive survey that has been conducted to find out users’ awareness regarding increased carbon footprint, requirement of features, and desire to reduce personal carbon footprint.

As part of our next work, we also present some important features and design challenges of a fuel efficient mobile GPS application for the iPhone platform that is able to suggest the “greenest” route to the user.

The rest of the paper is organized as follows. We discuss our motivation in the next section. In Section 3, we describe the existing methods of ubiquitous data collection and the challenges that those methods face. We focus on relevant related works in Section 4. We present the definition and properties of a ubiquitous carbon footprint in Section 5. The working principle of OCFF is presented in Section 6. In Section 7, we discuss the details of the OCFF platform. Design and implementation details of UCFC application are presented in Section 8. Results of user evaluation and implementation details with some screenshots are described in Section 9. In Section 10, we discuss some future works related to the mobile GPS application that is able to suggest most fuel efficient route to the user. Finally, in Section 11 we conclude the paper with some future research directions.

2. Motivation

Recently a growing number of companies have started to collect the carbon emissions data that result from their production and service provision. GHG (greenhouse gas) or carbon dioxide accounting methods may not only be applied to countries or organizations but also to individual persons. But, there are two main issues with calculators that track individual carbon emissions. First, while the existence of the calculators brings awareness to the issue of carbon emissions, they produce vastly inconsistent results given similar inputs. Second, these calculators are static and fail to take into account the dynamic behavior of human nature. In other words, these calculators need to respond to the current activities of an individual. Pervasive computing and sensor technologies offer great potential to estimate instant and ubiquitous CO_2 consumption of each individual or an organization as a whole. The use of carbon footprint calculator applications will increase the awareness of individuals to reduce their own CO_2 contribution to the environment. The use of ubiquitous technology to reduce CO_2 consumption can be better understood by the following examples.

2.1. Example 1

Rachel is a graduate student who spends most of her time working in her advisor's lab. This lab is a smart lab that has number of sensors installed within and outside of the room. The heating condition of the lab is preset so that it is comfortable for at most 10 people working at a time. However, when Rachel is working alone she can reduce her carbon footprint by increasing the room's temperature. Rachel's Smartphone has built in temperature and humidity sensors. The use of these sensor data along with the data collected from the room's outside temperature and humidity sensors can be used to determine her carbon footprint. A computational model can be used to estimate a carbon footprint at a specific time. This model may also compare her current carbon footprint with the average carbon footprint corresponding to one person (for example, using previous history to figure out the average carbon footprint for one person in a similar context). Finally, the application on Rachel's smartphone can display her current carbon footprint. The application can also provide a suggestion to reduce her carbon footprint by adjusting the room's temperature by some degree (for example, increase room temperature by two degrees). We name the application that can perform the above mentioned tasks “UCFC” application.

2.2. Example 2

Bob is a sales person who needs to go to different places everyday because of the nature of his job. He uses his GPS to determine the route to his destination. However, by using a simple smartphone application, Bob can reduce his carbon footprint markedly. Using the GPS of his smartphone, the application can track his travel and build a velocity profile-figuring out how efficiently he is driving. The application allows him to find the most fuel efficient routes for his vehicle between arbitrary end-points. As a result, it helps Bob to reduce his carbon footprint while he is driving. The application lets Bob review his trips with details that allow him to understand his efforts to reduce his carbon footprint better. Additionally, the application allows Bob to join an online carpool community. It can also show the results of people in his local area traveling in the same direction. This helps Bob to participate in the carpool and reduce his own carbon footprint even further.

In a later section of the paper, we will describe how we developed the UCFC application. We will also discuss how we will develop a small prototype of an application that can realize the scenario of example 2. In order to develop both of these applications, we need to collect several human behavioral data and environmental data for carbon footprint estimation. However, there are a number of research challenges for dynamic data collection that we will discuss in next section.

3. Ubiquitous data collection techniques and challenges

The usability of an automated carbon footprint calculator application is dependent on the data collection methods as well as on the collected data itself. The advances in mobile technology provide a unique opportunity to perform dynamic data collection that allows individuals to make environmentally responsible decisions. There are different techniques for obtaining relevant condition monitoring information that may be used in the estimation of a carbon footprint. Sensors can either be simply associated directly with the items of interest or it can simply report back any environmental situation. There is also a wide range of sensing options available from simple disposable TTI (time temperature indicators) that require a visual inspection to data loggers having either a fixed wired connection or using a wireless link, such as IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (used by ZigBee) or IEEE 802.11 (Wi-Fi) to transfer data to a datastore or real-time monitoring system. There are also some newer technologies such as: WISP (wireless identification and sensing platform) [5], [20] that can be used to collect environmental or human activity data. The GPS chip of mobile devices can also be used to track velocity of car or human location. Moreover, sensor-enabled RFID technology is also a good technique to track or trace through the monitoring of conditions such as temperature, contamination, shock, humidity, etc.

With current technologies, the commercial justification for using sensor enabled RFID tags is no longer limited by traditional constraints such as size and price, and they are now commercially available as active or semi-passive tags. Passive versions can be anticipated in the future if the power consumption of the sensing circuitry becomes sufficiently low or non-electrical sensors are used. Moreover, some important sensing capabilities of recent smart mobile devices can be used for different monitoring purposes. In the context of sensing and environmental or behavioral data collection, currently available technologies can be classified into two broad categories which are briefly discussed next.

3.1. Continuous monitoring

A continuous monitoring sensor provides time-related data samples that can be recorded at known intervals. Since the data can be collected at either regular or irregular intervals, any exceptional instances that occur during the course of the monitoring process can be stored. These sensors are ideally suited to goods that can degrade over time (e.g., as a result of temperature, humidity, etc.). Continuous monitoring has its pros and cons. On the positive side, the rich information is useful for evaluating the impact of condition changes in key processes for more detailed analysis. However, on the negative side, continuous data generally requires a power source, extra memory and additional components, which add to the cost, reliability, life and size of the tag.

3.2. Discrete monitoring

A discrete monitoring sensor provides a single binary state regarding the condition characteristic of an item or an event. For example, it detects whether the item has been subjected to any discrete event such as an occurrence of a damaging shock that exceeds acceptable limits. Discrete monitoring sensors have their advantages and disadvantages too. On one hand, the sensor does not require additional memory space due to the “true/false” binary characteristic, making it a cheap and simple solution. On the other hand, since there might not be any timing information as typically associated with continuous data collection, tracking down the origin of the event might be limited to corresponding object read points, provided that the state of the sensor was also interrogated at each read point.

The data collected using either discrete process or continuous process in a carbon footprint application has some important characteristics. For example, the aggregation of collected data over time for different parameters may infer a specific environmental situation (e.g., increase of temperature due to heating process). Even in some cases, the continuous change of data for a single parameter may be used to infer a specific situation (e.g., data collected from GPS of a mobile device can detect the change of velocity of its owner, which is an indication that the owner is driving a car or is in transport). Therefore, the nature of the collected data and their associated characteristics are not generalized. Moreover, generalization is complicated by the fact that in high-dimensional data sets, one size does not fit all. Hence, for example, developing a single regression model to represent all data is highly suboptimal. Hence, one option is to use data collected by a smaller population to build models capable of predicting the original characteristics of a larger population [13]. It solves a key problem at a critical phase of most newly deployed systems, which makes it important. Therefore, the correctness of any mathematical method or computational technique is dependent on appropriately modeling the nature of the dataset, which we will discuss in next section.

3.3. Computational model discrepancies

To measure personal or household carbon dioxide emissions, there are already plenty of carbon footprint calculators available on the Internet. Most of these calculators use quantitative models to estimate carbon emissions caused by the user's activities. However, each of the different carbon footprint calculator applications uses different computational models to estimate carbon footprints. For example, a computational model estimating the carbon footprint of a small office or room is different than the computational model that estimates the most fuel efficient route (resulting in less carbon footprint of a person) while driving. This difference in underlying computational models is attributable to the sparse nature of the characteristics of the data. Since data collected in different carbon footprint applications are different, the computational model using those data will also have to be different for different applications. Admittedly, it is difficult to find a general mathematical or computational model that can be used to estimate carbon footprints in all applications. Therefore, it is wise to use a single platform for developing different carbon footprint calculator applications that will use different computational model for different applications. The computational model component of the platform can be allowed to be customized by the developers.

In this paper, we are addressing the above mentioned research problem by proposing a novel framework for developing different ubiquitous carbon footprint calculator applications. Our framework has a computational model component that can be customized and adapted for different applications. For example, UCFC application and the application of example 2 (mentioned in Section 2) use different mathematical models for estimating carbon footprint corresponding to two different problem scenarios. Both of these applications use OCFF as the development framework and they use different mathematical models for CO_2 estimation.

4. Related work

Recently new environment related software products emerged to reduce energy consumption and the number of these products is still increasing on daily basis [11]. In [17], a green software infrastructure has been proposed to help non-professionals make small contributions to address the climate change problem. Darby provides a detailed survey of studies on electricity feedback systems from the past three decades [25]. One other feedback system is the carbon footprint calculators that are available these days. They are available from two types of sources. One source is the Internet. There are lots of free carbon footprint calculators available on the web that are free of cost. Different companies and organizations develop other types of calculators.

The recent rise in footprint calculators has been accompanied by inconsistencies in output values given similar inputs for individual behavior [18]. In some cases, values can vary by as much as several metric tons per activity. These variations in output can influence both the types of steps individuals take and the overall level of effort to reduce contribution in climate change.

In this section, we discuss some of the widely know carbon calculators. Average values of household electricity consumed are provided optionally by five calculators: American Forests [1], BEF [4], The Conservation Fund [8], EPA [12], and TerraPass [23], [24]. In general, the amount of electricity consumed is converted into CO_2 emissions using conversion factors. Most conversion factors are based on U.S. national averages; however, given the geographic differences in CO_2 levels associated with methods of electricity generation, Be Green [3], BEF [4], Safe-Climate [19], and TerraPass [23], [25] use conversion factors that are state-dependent. All these calculators discussed so far are static calculators that determine users' carbon footprints based on the data provided manually. In this paper, we propose a carbon footprint calculation framework that allows developers to develop applications taking into account the dynamic activity of user.

5. Definition and properties of ubiquitous carbon footprint

A carbon footprint is the total amount of greenhouse gas (GHG) emissions caused by an organization, event, or product. However, in this paper, we introduce the notion of a *Ubiquitous Carbon Footprint* (*UCF*) that can be addressed through several quantifiable *context contributors*. By context contributors, we refer to those dynamic human activity or environmental changes that are responsible for significant CO_2 emissions. UCF consists of various context contributors and most of them evolve to address the unique characteristics found in ubiquitous computing environments. Thus, we define the properties of UCF, in this paper, from their scope in ubiquitous scenarios. In our model, total ubiquitous carbon footprint UCF_t depends on the number of context contributors (cc_1, cc_2, …, cc_k) at time t. We consider the following notations:

(1) $UCF_t(cc) = \sum_{i=1}^{k} UCF_t(cc_i)$

here, k = number of context contributors.

Context contributors can be categorized in two types: *explicit context contributor* (*ecc*) and *implicit context contributor* (*icc*). Explicit context contributors are those contexts that directly contribute towards CO_2 emission (e.g., burning fuel, electricity consumption). Implicit context contributors depend on the *ecc* to contribute in CO_2 emission. Using an air conditioner or heater, increasing or decreasing room temperature are examples of *icc*. Therefore, we can redefine Eq. (1) as:

(2) $UCF_t(cc) = UCF_t(ecc) + UCF_t(icc)$

where $UCF_t(ecc) = \sum_{i=1}^{m} UCF_t(ecc_i), UCF_t(icc) = \sum_{j=1}^{m} UCF_t(icc_j)$; here, m = number of explicit context contributors and n = number of implicit context contributors.

Details of the equation for the overall ubiquitous carbon footprint calculation are described in later sections of the paper. Throughout this paper, we use the term "mobile device" to refer to a "person" as we assume that each individual wishing to determine her personal carbon footprint will have a mobile device (cell phone, PDA, smart phone, etc.). A UCF has following properties and relationships:

1. *Context dependant*: A mobile device will have different carbon footprint values for different contexts.

$$UCF_t(cc_i) \neq UCF(cc_j) \text{where}, i \neq j$$

However, the carbon footprint value of some aggregated contexts may be similar to that of a single contexts or some other combination of aggregated contexts. For example,

$$UCF_t(cc_m) + UCF_t(cc_n) = UCF_t(cc_i) = UCF_t(cc_x) + UCF_t(cc_y) + UCF_t(cc_z)$$

2. *Dynamic*: A mobile device may have different carbon footprint values at different times provided that the context contributor has been changed.

$$UCF_{t+1}(cc_i) \neq UCF_{t+2}(cc_j),\text{where} i \neq j$$

However, if the context contributor is same over a period of time, carbon footprint can be same.

$$UCF_{t+1}(cc_i) = UCF_{t+2}(cc_i)$$

3. *Partially-transitive*: If device A's carbon footprint depends on a particular *icc*, which is dependent on another *ecc*, then this chain relation between *icc* and *ecc* implies that a carbon footprint value can be inferred for A on *icc*.
4. *Non-symmetric*: In general, two devices A and B having the same carbon footprint value do not necessarily indicate that both devices depend on the same context contributor. Let $UCF_t(cc_i)$ be the carbon footprint of device A and $UCF_t(cc_i)$ be the carbon footprint of device B. Therefore,

$$UCF_t(cc_i) = UCF_t(cc_j) \Rightarrow cc_i \neq cc_j$$

5. *Context relation dependent*: UCF is dependent on relations among different context contributors.

6. Working principle of OCFF

OCFF aims to provide a ubiquitous carbon footprint calculation and carbon footprint abatement consulting service for individual users. With the sensors installed inside the portable devices, the system can collect the information needed for pervasive CO_2 emissions calculations. OCFF is supported by a calculation platform which contains quantitative computation models of human–environment interaction. Experts can use collective intelligence to update the models and offset knowledge on this platform. In order to develop a flexible carbon calculation platform, we need to establish and manage a set of flexible human–environment interaction models and to find a solution for model integration. The result obtained from an individual user's impact on the environment can then be used to match the relevant knowledge and suggestions in the knowledgebase, and consequently, promote awareness of reducing carbon emission and facilitate GHG offset knowledge sharing among individual. Component wise our system has two major parts:

6.1. Portable devices component

It finds major parameters in the human behavior model. It finds data collection methods from sensors and portable devices.

6.2. Model management and calculation component

This part finds a way of managing a set of quantitative models for calculating personal carbon footprint. It can build up a customized carbon footprint calculator using applicable quantitative models.

Fig. 2 shows the conceptual framework of a ubiquitous carbon footprint calculator. The figure illustrates the example of working principle of OCFF. Let us consider the scenario illustrated in Fig. 1. This scenario corresponds to the UCFC application mentioned in Section 2. In this scenario, a user uses a handheld device which is embedded with several sensors. The mobile device can communicate with the sensors installed in a room and outside a room. For example, the mobile device can measure the inside temperature of a room, and it can collect the outside temperature from the sensor installed outside the room.

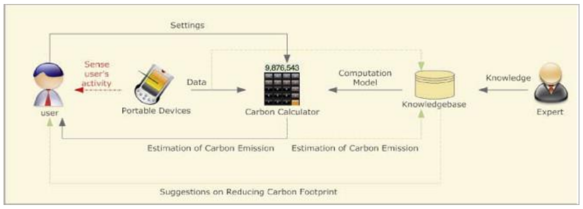


Fig. 1. Example of UCFC application for individual users based on OCFF.

UCFC can take the difference of average inside room temperature and outside temperature. This temperature difference refers to the increase of temperature that caused by air conditioning. And the power consumption of that air conditioning will result in GHG emissions. By using proper quantitative models, we could estimate the average increase of carbon footprint for each individual person of that room. Our system could provide appropriate suggestion (shown in Fig. 2) to user according to the estimation that helps them to minimize their carbon footprint.

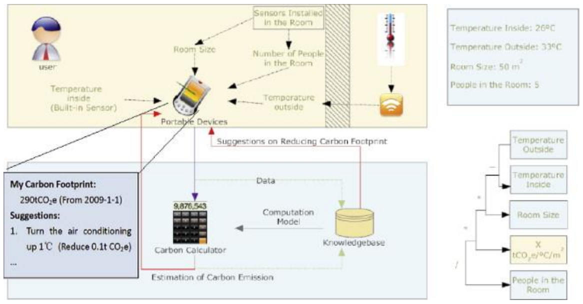


Fig. 2. Illustration of UCFC application where suggestion is displayed in user's mobile device.

7. Model management and calculation of OCFF

In our practice, the quantitative models used for the calculation carbon footprints could be local computation functions or web services provided by a third-party. To build the model registry, we use a decomposition tree structure to store the models' information. The root node of the tree structure represents the goal of calculating personal carbon footprint, while its descendant tree nodes represent the specific calculation goals in specific areas decomposed hierarchally from the top goal. Next, we will discuss the layered architecture structure of OCFF, and then we will discuss the details of our calculation model.

7.1. Architecture of OCFF

In this section, we are going to introduce our approach of building a flexible ubiquitous personal carbon footprint calculation platform. The platform can manage a set of simple footprint calculation models and can integrate them to calculate the overall carbon footprint of an individual user.

Fig. 3 shows the system architecture of the personal carbon footprint calculation platform. The system includes three layers: the *Model Layer*, the *Middleware Layer* and the *Sensor Layer*. The Sensor Layer collects raw data from the registered sensors and smart meters and sends the data to the Middle Ware Layer. Raw data is then analyzed by the middleware to remove noise and recognize some current conditions of the user (for instance whether he is currently traveling by car or by bike). When the middleware layer finishes analyzing the data, the processed data along with the deducted current conditions of the user serves as the model input for the Model Layer for footprint calculation.

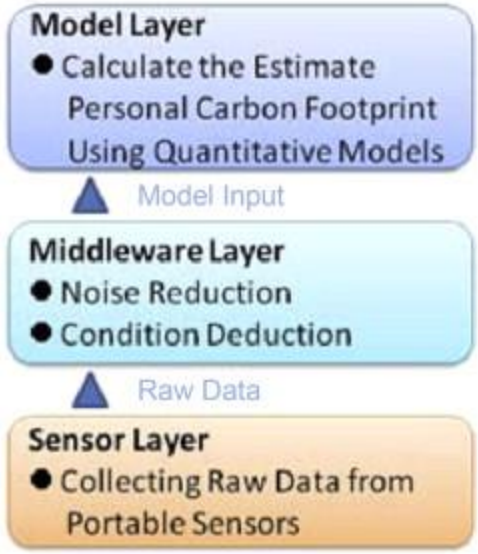

Model Layer
● Calculate the Estimate Personal Carbon Footprint Using Quantitative Models
Model Input
Middleware Layer
● Noise Reduction
● Condition Deduction
Raw Data
Sensor Layer
● Collecting Raw Data from Portable Sensors

Fig. 3. Data flow in different layers of OCFF.

The basic architecture of the OCFF framework is shown in Fig. 4. OCFF is designed in such a way so that it can be modified or customized by application developers. We designed the architecture of OCFF to facilitate the application developer in obtaining all kinds of core services to develop her application in semi-transparent manner.

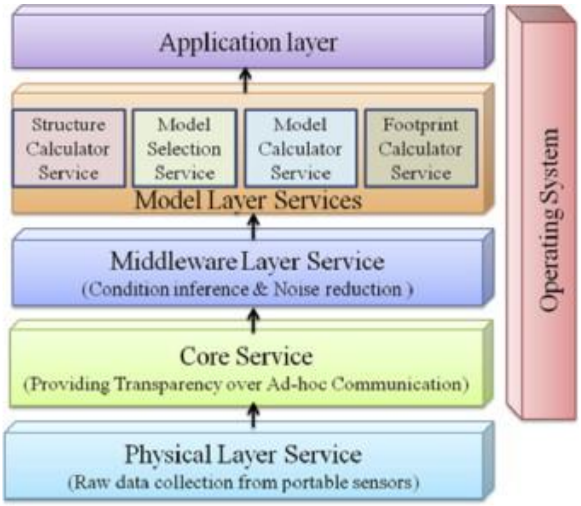

Application layer
Structure Calculator Service
Model Selection Service
Model Calculator Service
Footprint Calculator Service
Model Layer Services
Middleware Layer Service
(Condition inference & Noise reduction)
Core Service
(Providing Transparency over Ad-hoc Communication)
Physical Layer Service
(Raw data collection from portable sensors)
Operating System

Fig. 4. Basic architecture of OCFF framework.

7.2. Calculation models in OCFF

The OCFF framework uses a set of quantitative models to calculate the user's personal carbon footprint. All of these quantitative models get their input data from sensors and their respective middleware. All models have a GHG emission estimation value as their calculation output. Each of the models could calculate a certain category of the user's carbon footprint under some restraints.

Just as Fig. 5 shows, the system uses a three attribute structure to describe a model, representing its requested inputs for calculation, calculation restraints and the model's error. The restraints attribute of a model can also be divided into general restraints and trigger restraints: the general restraints show under what circumstances the model is capable of doing the calculation. The trigger restraints, on the other hand, represents when the model should be activated for calculation.

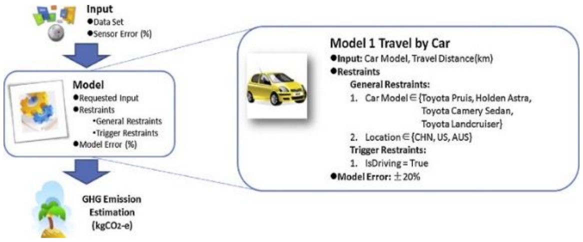


Fig. 5. Personal carbon footprint calculation using a *Single Quantitative Model*.

Take the example of Model 1 shown in Fig. 6, according to the general restraints, the model could calculate the user's estimated carbon emission for a car with the error less than 20%, if the user is driving one of the listed four kind of cars and is traveling in China, the United States or Australia. The trigger restraints show if the data could determine that user is driving and all the general restraints are satisfied, then the model should be used in the calculation. The calculation error for a single model could be estimated using the sensor error and the model error. This calculation error could later be used to choose models for the individual user from the entire model set, and to calculate the system's overall calculation error.

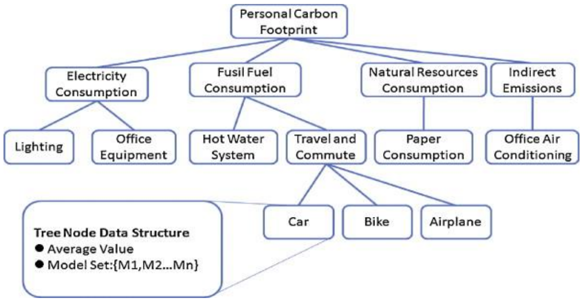


Fig. 6. An example hierarchal model collection tree.

7.3. Model storage structure: hierarchal model collection tree

After learning about the representation of a single model, the following questions still lie ahead: using what kind of data structure can we store all these models? How can these models work together to calculate the overall carbon footprint for personal users? For the model storage structure, there are concerns of whether it is easy to update and search and whether it is effective for model selection and calculation. We suggest a decomposition tree structure to store the models. Fig. 6 shows an example of the Hierarchal model collection tree (HMCT) in which we store the quantitative models used for calculation. The HMCT is a decomposition structure of the personal user's carbon footprint with the root node representing user's overall carbon emission. The root node and its descendent nodes can be divided into several smaller subcategories each forming a sub-node of the current node.

In the example shown in Fig. 7, a user's personal carbon footprint is divided into direct carbon emission from electricity consumption, fossil fuel consumption, natural resources consumption, and other indirect emissions. And each of the tree nodes could also be divided into even smaller categories, like the electricity consumption node could be divided into two sub-categories: direct carbon emission from electricity consumption caused by lighting and direct carbon emission from electricity consumption caused by using office equipment.

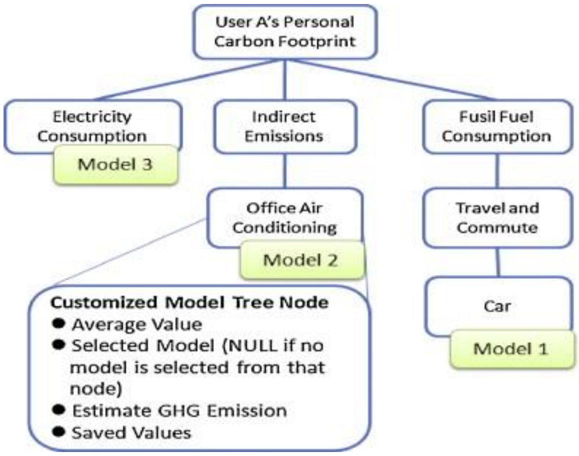


Fig. 7. An example Customized Model Tree.

Each tree node of the HMCT has two attributes: one is the actual measured average carbon emission value of that sub-category that the node represents, and the other is the model collection of all the models which fall to the sub-category that tree node represents.

The HMCT does not need to be exhaustive at first. As the new models, which could not be covered by the existing tree node specified categories, are introduced to the system, new tree nodes will be added to comply with that change. The HMCT will be managed manually by domain experts.

7.4. Model selection for individual user in OCFF

When the user is defined, according to the data availability and model restraints, a set of models along with the sub-structure of the Model Collection Tree can be selected to form a Customized Model Tree. The Customized Model Tree is used not only for storing the calculation models for specific user, but also for the calculation of the overall carbon emissions for the user. To avoid duplicate calculations, we select the model according to the following rules:

- At most one model can be selected from a certain HMCT tree node's model set.
- If one model is selected from a certain HMCT tree node, no model could be selected from its descendent nodes.

Under such rules, there are usually more than one set of models that could be selected from the HMCT. So there are several strategies that could be applied during selection according to the user's preference. For example, the user could select the model set with the Minimum Calculation Error or the Maximum Model Coverage. We use the Minimum Error Strategy, which uses a Depth First Search to select a set of models with the lowest calculation error. After the model set is selected from the HMCT, the Customized Model Tree also keeps the sub-structure of the HMCT from the tree root to the nodes with selected models.

Although the Customized Model Tree is much like a HMCT after branch cutting it has a slightly different data structure for tree nodes. Tree node in the Customized Model Tree has four attributes: average value is the same as the one in the HMCT tree nodes; selected model is the model being selected from its co-responding HMCT tree node (NULL if there is no model selected); estimated GHG emission, which represents the accumulative total carbon emission calculation result from the branch formed by the node and its descendent nodes; and saved values is a set of constant values collected from the user which will be used as part of the calculation input for selected model in the node.

Fig. 8 shows an example of a typical Customized Model Tree. According to user A's given data and sensors that user A can access, three models, out of all the models whose restraints could be satisfied, can be selected from the HMCT according to the above mentioned method. The Customized Model Tree could then be built on basis of the selected models and the sub-structure of the HMCT. For the node of the Customized Model Tree, node Electricity Consumption, Office Air Conditioning and Car have the selected model saved in their selected model attribute, respectively. From the user input data, the car model and the location of user A's house and office are then saved in the saved values attribute of the three tree nodes. When the node first appears in the user's Customized Model Tree, the estimate GHG emission will be set to 0 kg CO_2-e.

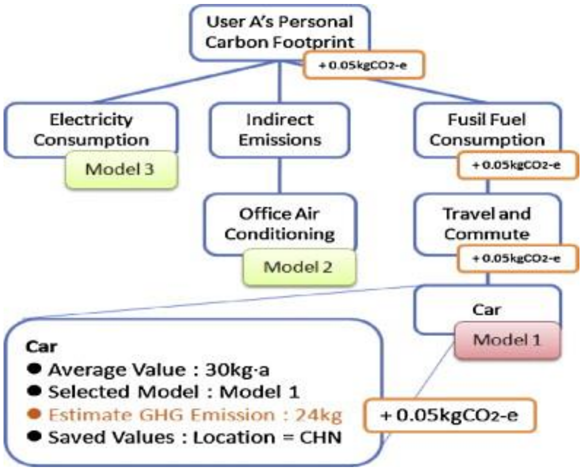


Fig. 8. Personal carbon footprint calculation.

7.5. Personal carbon footprint calculation in OCFF

After the Customized Model Tree is constructed, the user's personal carbon footprint can be calculated using the tree. The system will calculate this using the following five steps:

1. Building a multi-dimensional index tree using the trigger restraints of the selected models.
2. Collecting data automatically from the data sources according to the preset time step.
3. When the data is collected, the system will check the trigger restraints index tree and find the active models in the Customized Model Tree.
4. For each active model, it will calculate the estimated GHG emissions increase according to the input.

The increase of carbon emission will then be added to the estimated GHG emission attribute of the active model's node and also to the node's ancestor nodes in the Customized Model Tree.

In Fig. 9, an example of calculation is given to illustrate this process. After the trigger restraints index tree being built and a set of input data being collected, the system checks the index tree and finds that the trigger restraints for Model 1 is satisfied. After Model 1s general restraints being checked, Model 1 is activated, an increase of carbon emission can be calculated using the input data and the saved values of the node car. The calculation results in an increase of 0.05 kg CO_2-e. This increase is added to the estimated GHG emissions attribute of car and its ancestors.

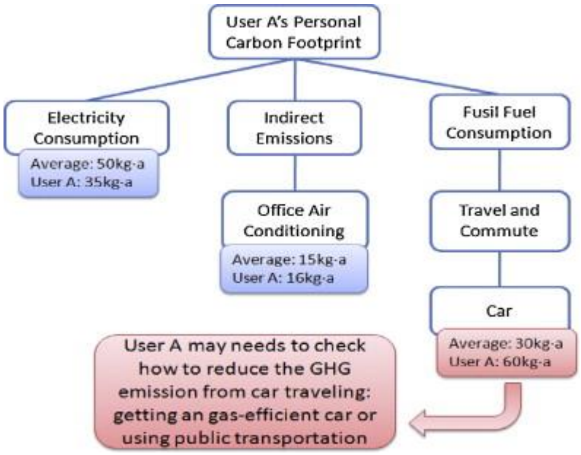


Fig. 9. Calculation result analysis.

Besides simple calculation, the system can also help the user to identify the important emission sources. As Fig. 9 shows, by comparing the speed of increase for the estimated GHG emissions with the average value of the Customized Model Tree nodes, user A can notice that he has emitted more carbon dioxide than average in Car Traveling. He can use the sub-category of the node as an index to search for the tips for reducing CO_2 from the knowledgebase. So far we discussed the model management and calculation method of OCFF. Next we will discuss the design and implementation of UCFC application based on OCFF.

8. Design and implementation of UCFC application

Our proposed carbon footprint calculator is different from the currently existing ones. Our system can use the built-in sensors of the portable devices as a basic source of input. According to our investigation, there are two types of footprints, each of which is shown in Fig. 10. We focus on determining the carbon footprint of a user for his/her ubiquitous activity. In this section, we discuss the details of system diagram, design and implementation of UCFC application. We present the system diagrams of UCFC. Then we will briefly discuss some of the design diagrams of our system. Finally we will look into the details of hardware implementation.

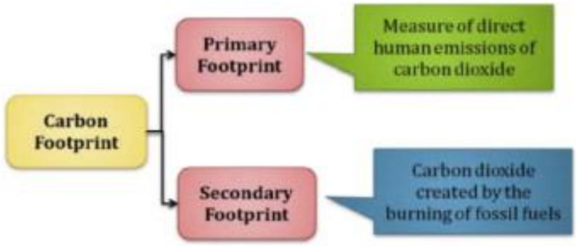

Carbon Footprint
Primary Footprint
Measure of direct human emissions of carbon dioxide
Secondary Footprint
Carbon dioxide created by the burning of fossil fuels

Fig. 10. Two types of carbon footprint.

8.1. System diagram of UCFC application

Carbon dioxide calculators generally work by accepting user inputs characteristic of an individual's behavior and returning an amount of carbon dioxide emitted as a direct result of such behavior in the form of a user's carbon footprint. This section investigates the dynamic behavior of human activity and tries to incorporate those factors in the calculation of a carbon footprint. It is different from previous research on carbon footprint calculation as it is a *dynamic* carbon footprint calculator. UCFC has three main components which are shown in Fig. 11.

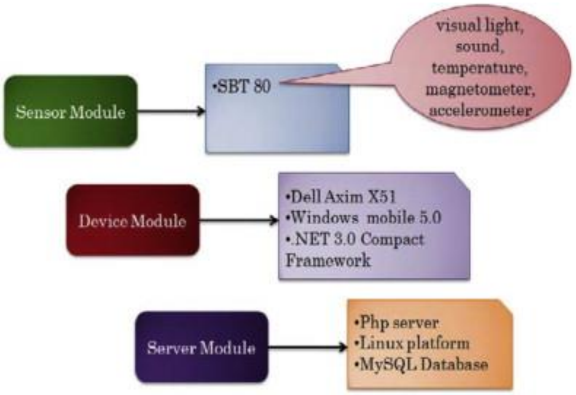


Fig. 11. Main components of UCFC application.

The system diagram of UCFC is shown in Fig. 12. Our system can work in an ad hoc manner. We divided the entire system in three different modules/components. User of our system only needs to have a mobile phone and the simple sensor module to be able to use the carbon footprint calculator. Next we discuss the details of each of the components of UCFC application.

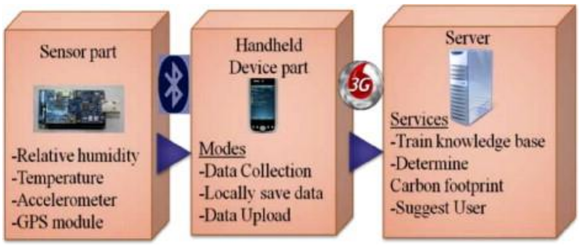


Fig. 12. System diagram of UCFC.

8.1.1. Sensor module

Our sensor module performs people-centric sensing to find the amount of carbon dioxide emissions. The sensor module consists of small, low powered, and very low cost sensors. The main objective of the sensor module is to collect environmental and human-behavioral data that can inform the calculation of the carbon footprint of a person. The sensor module also collects accurate data from a variety of calibrated sensors over time. Another task is to filter raw data to select those that are useful for the calculation of carbon footprint. In our sensor module, we have used a GPS sensor to collect the position of the user. The sensor module has a Tmote that collects GPS data from a GPS module and other sensor data collected from the sensor board EasySen SBT80. The EasySen SBT80 sensor board is used to collect several environmental data such as temperature. The Tmote sends all these data to the intermediate handheld device using a Bluetooth module. In our system collected sensor data are also displayed in a webpage. Currently the collected sensor data are: (1) GPS data: latitude and longitude, (2) average inside room temperature, (3) inside humidity, (4) outside temperature, (5) outside humidity, (6) speed, and (7) acceleration. For our prototype system we are currently displaying only some of the sensor data (location, temperature, humidity, and time) in the device interface.

8.1.2. Handheld device module

Our handheld device module works in an ad hoc manner. We have collected data from various sensors we mentioned in sensor module. We have developed an application in Microsoft's. Net framework that can take input from the sensor module and show the output in the mobile device. This module talks with the Bluetooth module and collects sensor data via the Bluetooth module. It saves the collected sensor data temporarily in the mobile device. Finally it uploads locally saved data to a central server in the form of XML or web services for further processing in knowledge base. In the event that Internet connectivity is unavailable, the device can postpone uploading data to the server. Whenever, the device has a nearby WiFi or wireless access point, it can upload the data in the server. The device module can also upload data through the user's intervention.

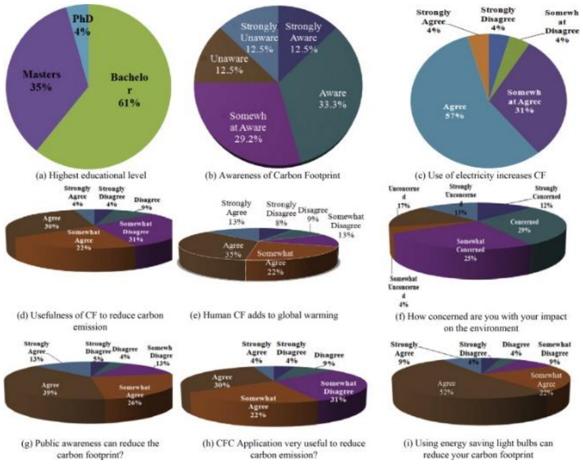
8.1.3. Server module

Our server module does the main task of calculating the carbon footprint for an individual user with the integrated quantitative model based on the perceived data from the handheld device. In the mean time, domain experts can also access the server module to manage the model storage models and update existing models. In our system, we have integrated the device module with the model management and calculation part using XML or web services. After the calculation of a personal carbon footprint, the server can search for appropriate suggestions for the user based on the result. This suggestion is sent to the device module together with the carbon footprint. This information will then be displayed in the user's mobile device application interface.

8.2. Software design and implementation of UCFC application

In this section we summarize our requirements collection process and analysis, design process, and design diagrams.

Pre-design survey: We conducted an extensive survey before the design of UCFC to perform requirement analysis. The survey was conducted with 30 people to understand their concerns regarding increased carbon footprints, their requirements, and finally their practices and desires to reduce personal carbon footprint. The survey results are shown in Fig. 13.



(a) Highest educational level

(b) Awareness of Carbon Footprint

(c) Use of electricity increases CF

(d) Usefulness of CF to reduce carbon emission

(e) Human CF adds to global warming

(f) How concerned are you with your impact on the environment

(g) Public awareness can reduce the carbon footprint?

(h) CFC Application very useful to reduce carbon emission?

(i) Using energy saving light bulbs can reduce your carbon footprint

Fig. 13. Pre-design survey results and finding.

In this figure, the survey result for each question is shown as a subfigure. The subfigure caption states the question corresponding to that result. Among the 30 participants of the survey some of them were computer engineers, some were from mathematics background, some from bioinformatics background. There were 61% with their highest educational qualification up to bachelor level, 35% having education qualification of masters and only 4% had qualification of PhD (shown in Fig. 13(a)). We deliberately chose individuals with diverse backgrounds involved in different types of pursuits in different types of environments, in order to compare and contrast the requirements and use of footprint calculator in a broad variety of situations. We used about 25 questions in the survey that were meant to gather information about people's tasks, the manner in which they use energy in their daily lives, their concern regarding the effect of increased CO_2, their requirements, expectations, and finally their desire to use a mobile carbon footprint calculator ubiquitously to participate in CO_2 reduction.

The motivation behind this survey was to incorporate the inference and findings of the survey results in the design and implementation of UCFC prototype application. Some of our inferred and analyzed survey results indicate that:

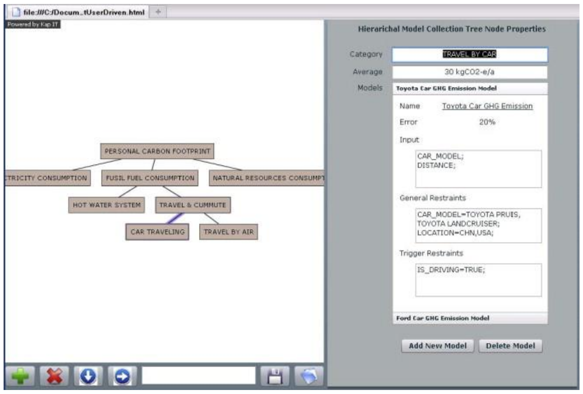
- Roughly two-thirds of those surveyed were familiar with the idea of carbon footprints.
- Roughly two-thirds of those surveyed were concerned about carbon footprint and energy usage.
- 17% of those surveyed had used or were planning to use a carbon footprint monitor
- Over half of those surveyed thought carbon calculators will be useful for reducing carbon emissions.
- Of those concerned about their environmental impact, the greatest limiting factor for reducing energy usage was the amount of effort required.

Some of the conclusions from survey results are:

> *Ease of use:* Data gathering and accuracy of carbon footprint are important. Equally important is convenience. If a monitor requires too much interaction, or is too complicated, few will use it. We need to design user interface with this in mind. We have to automate data collection as much as possible.
>
> *Audience:* We have to realize that a third of those surveyed were not concerned about carbon footprint impact. A large percentage of people will not be interested in the monitor at all. We further need to focus on creating a functional monitor and convince people to use it.

8.2.1. Software design of UCFC application

According to the model management and calculation method, we have developed a management tool for domain experts to manage the stored quantitative models. As Fig. 14 shows, the tool allows an expert to manage the tree structure and to add, delete or update the models stored in certain tree node.



Hierarichal Model Collection Tree Node Properties

Category: TRAVEL BY CAR
Average: 30 kgCO2-e/a
Models: Toyota Car GHG Emission Model

Name: Toyota Car GHG Emission
Error: 20%
Input:
CAR_MODEL;
DISTANCE;

General Restraints:
CAR_MODEL=TOYOTA PRUIS,
TOYOTA LANDCRUISER;
LOCATION=CHN,USA;

Trigger Restraints:
IS_DRIVING=TRUE;

Ford Car GHG Emission Model

Fig. 14. Interface for HMCT management tool.

8.2.2. Design diagram of UCFC application

Based on our analysis of the survey results, we identified some key tasks and goals that need to be satisfied by the UCFC application. As a first step towards developing UCFC application, we designed the use case and the class diagram of the system.

In the development of a demo system for the example mentioned in Section 2, we have the class diagram of the system for data collection and storage. As Fig. 16 shows, the system is capable of collecting and storing data about temperature, humidity, location, etc., from portable devices. The design is also open for future extension to include more kinds of sensors and smart meters.

These diagrams are designed keeping in mind the generalized nature of the application. For example, the class diagram shows certain specific sensor modules, but it can be extended further to include more sensors using the generalized interface of the system.

Fig. 15, Fig. 16 show the use case diagram and the class diagram of the system, respectively.

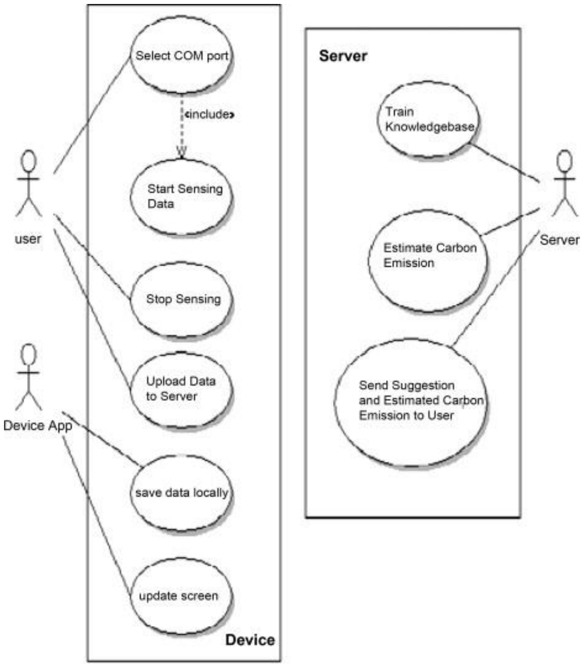


Fig. 15. Use case diagram of UCFC application.

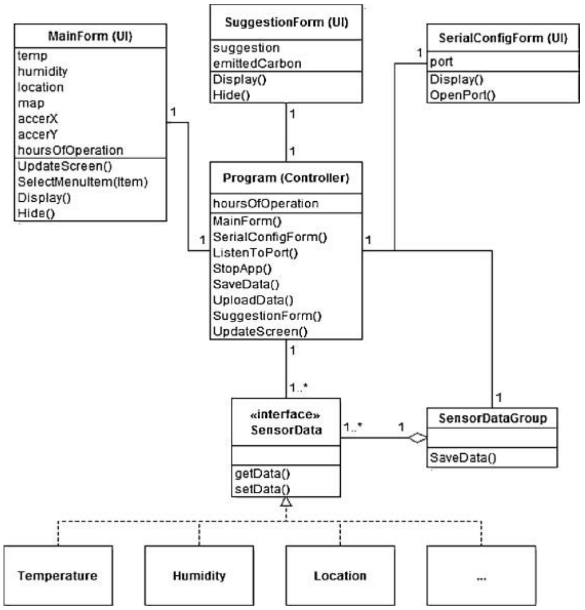


Fig. 16. Class diagram of UCFC application.

8.2.3. Hardware implementation of UCFC application

In the server module, we have created a MySQL database, carbon concentration (CCon) database. δ values for different states are stored in this database. δ values refer to the amount of carbon concentration per degree Celsius per meter square. δ values vary based on the location of the user. Therefore, for each state in United States, there exists a different δ value. These δ values are collected from [16].

An appropriate δ value is used in the equation by deriving the state/location of user from his/her GPS data. We have used PHPMyAdmin for data administration purposes. Here, the reference inside temperature is determined by taking the average of inside temperatures collected over a long period of time. If this reference temperature is greater than the current inside temperature, then the calculator can infer that the heater in the room is burning more carbon dioxide. The amount of determined carbon emissions is divided by the total number of people in the room so that an average individual carbon footprint can be measured. Fig. 17 shows the block diagram of the sensor network of UCFC system.

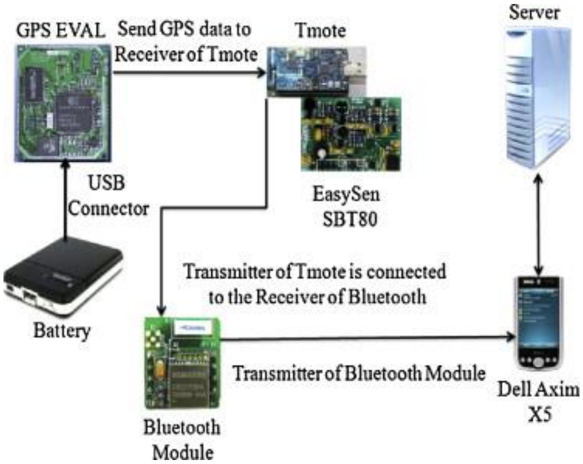

GPS EVAL
Send GPS data to Receiver of Tmote
Tmote
Server
USB Connector
EasySen SBT80
Transmitter of Tmote is connected to the Receiver of Bluetooth
Battery
Transmitter of Bluetooth Module
Dell Axim X5
Bluetooth Module

Fig. 17. Sensor network of UCFC application.

8.2.4. Example calculation model used in UCFC application

In this section we provide a quantitative model that is used for the calculation of the example scenario discussed in Section 4. We use a simple model to determine the carbon footprint from heating. Table 1 summarizes the notations used our model.

Table 1. Summary of notations.

Δ_t	Carbon footprint due to temperature increase/decrease
n	Number of parameters
δ	Average tons of CO_2/°C/m^2 in different states
RT	Reference inside temperature
IT	Current inside temperature
RS	Room size
N	Number of people in room
$UCF_t(cc)$	Ubiquitous carbon footprint at time t for context contributor cc

Model description:

Input: indoor temperature, outdoor temperature, room size, number of people in the room
General restraints: Location ∈{USA}
Trigger restraints:
|indoor temperature − outdoor temperature| > 5 °C
Error: 40%

Equations used in the model:

In this scenario, context contributor

cc = temperature change

$$\Delta t = \frac{\{(RT - IT) \times RS\}}{N}$$

$$UCF_t(cc) = \Delta_t \times \delta t/°\mathrm{C/m}^2$$

9. Evaluation of UCFC application

We evaluated our UCFC prototype application in the following ways:

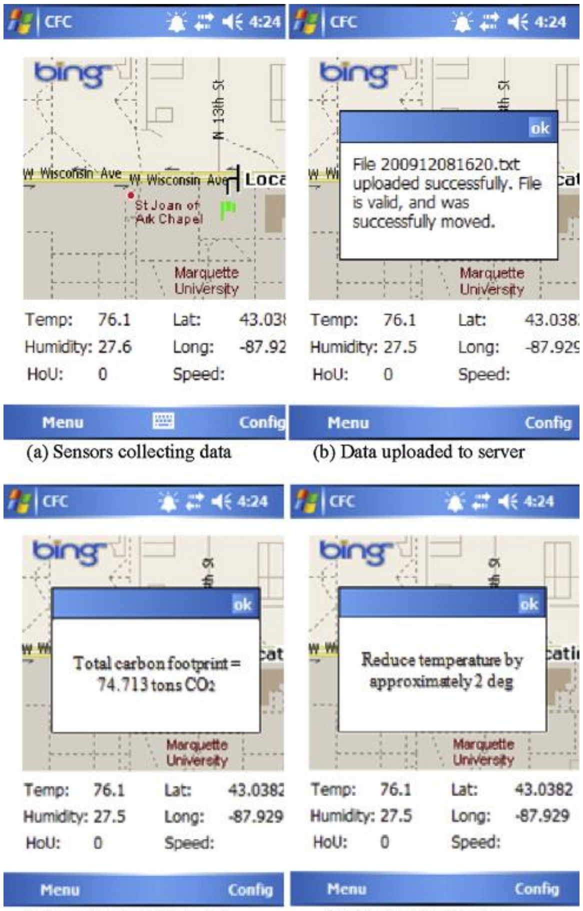
1. Prototype implementation
2. Quality of service measurement
3. Cognitive walkthrough strategy

9.1. Prototype implementation

Table 2 summarizes the implementation details. We have completed the first prototype of UCFC. In Fig. 18 we provide some of the screen shots of our implementation. At the beginning the interface is displayed to the user once she starts the UCFC application. Next she selects the appropriate COM port. As soon as she selects correct port, the sensors start to collect the data (shown in Fig. 18(a)).

Table 2. Implementation details.

Operating system	WINCE
PDA hardware platform	Dell Axim X51v
Programming language	VC# .Net compact framework
Mobile ad hoc mode	IEEE 802.11b
Server language	PHP
Database	MySQL
Sensor to mobile comm.	Bluetooth
Tmote/sensor OS	TinyOS



Temp: 76.1 Lat: 43.038
Humidity: 27.6 Long: -87.92
HoU: 0 Speed:

(a) Sensors collecting data

File 200912081620.txt uploaded successfully. File is valid, and was successfully moved.

Temp: 76.1 Lat: 43.038
Humidity: 27.5 Long: -87.929
HoU: 0 Speed:

(b) Data uploaded to server

Total carbon footprint = 74.713 tons CO2

Temp: 76.1 Lat: 43.0382
Humidity: 27.5 Long: -87.929
HoU: 0 Speed:

(c) Total carbon footprint

Reduce temperature by approximately 2 deg

Temp: 76.1 Lat: 43.0382
Humidity: 27.5 Long: -87.929
HoU: 0 Speed:

(d) Suggestion for user

Fig. 18. Screenshots of the prototype implementation of UCFC application.

After sometime, the user stops the application and uploads the locally saved data file to the server using the Bluetooth module. If the file is transferred to the server successfully, a message is displayed to the user, which is shown in Fig. 18(b). Then the carbon footprint of the user is displayed to the user (Fig. 18(c)). Finally, a suggestion is given to the user so that she can reduce her carbon footprint, which is shown in Fig. 18(d).

9.2. Quality of service measurement

We also evaluate our system by determining the quality of service of our calculation. Since the user may also be interested in the quality of the calculation result. In the system, two parameters are named to represent the quality of the overall result for a chosen Customized Model Tree: overall error and model coverage.

For the system calculation error, a single model its calculation error can be calculated by following formula:

$$\text{single model calculation error} = (\text{sensor error} + 100\%)(\text{model error} + 100\%) - 100\%$$

Because the final calculation result is the sum of all the select model results, the overall calculation error should not exceed the largest calculation error of the select models. We can use the following formula to estimate the overall error of the system:

$$\text{overall calculation error} \leq Max\{(\text{sensor error}i + 100\%)(\text{model error}i + 100\%) - 100\% | \text{Model}i \in \text{selected model set}\}$$

By using this formula, the system can apply certain strategies during the building of the Customized Model Tree to lower the system's overall calculation error.

In addition to system error, a user may also be interested in the completeness of the calculation. In the ideal condition, most of the user's activity would be captured by the sensors and sent to the system for calculation. But it is hardly so in the real world. Because of the lack of input data, only a small proportion of the user's carbon emission activities can be recognized and calculated.

To explain how complete the calculation result can be, in our approach, we assume the HMCT has a rather complete view of the user's carbon footprint. We use a model coverage rate to evaluate to what extent the system can capture the user's carbon emission. The Model Coverage Rate of a Customized Model Tree can be calculated with the following steps:

Assume the tree root node of the Model Collection Tree has the

$$Coverage\ Value\ of\ 100\%:$$

$$CoverageValue_{root} = 100\%$$

For each node *k* in the HMCT:

$$CoverageValue_k = \frac{CoverageValue_{Parent(k)}}{NumberofChildren(Parent(k))}$$

The Model Coverage Rate of a Customized Model Tree can be calculated using the Coverage Value of its co-responding nodes in the HMCT:

$$ModelCoverageRate = Sum\{CoverageValue_i | Nodei \in CustomizedModelTree\}$$

Fig. 19 shows an example of the Model Coverage Rate calculation. In our example, the root node has the coverage value of 100%, and its direct descendant nodes all have the coverage value of 25%. Because the node “Office Air Conditioning” is the only descendant of its parent node, the coverage value for it is 25%. The node “Car”, in which Model 1 lies, has a coverage value of 4.17%, one third of that of its parent node, and one sixth of that of its grandparent node. As the system selected Models 1, 2, and 3 to form the Customized Model Tree, the model coverage rate of the selection could be calculated:

$$ModelCoverageRate = 25\% + 25\% + 4.17\% = 54.17\%.$$

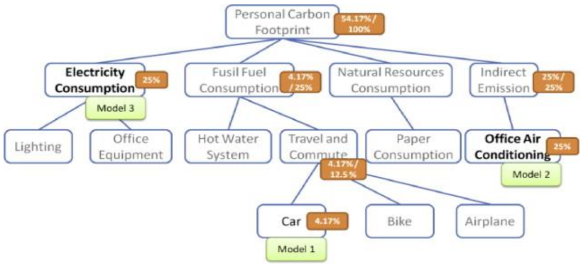
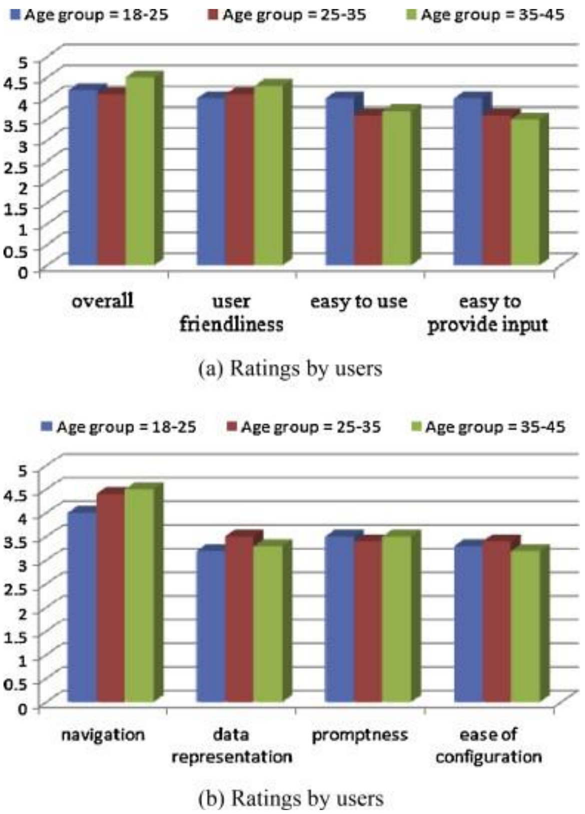


Fig. 19. Calculating model coverage for *Customized Model Tree*.

The result explains that approximately half of the user's activity could be captured and calculated by the system. We are in the process of incorporating this quality of service measurement in our developed prototype. Our idea is to display a message to the user stating the quality of the overall carbon footprint calculation. This technique would also develop the users’ reliability on UCFC application.

9.3. User evaluation with cognitive walkthrough strategy

The user experience and opinion of the UCFC application has been examined by means of a cognitive walkthrough among people from various age groups, diverse backgrounds and different degrees educational attainment. This strategy [7] encompasses one or a group of evaluators who inspect a user interface by going through a set of tasks and assess its understandability and ease of learning. The survey included 30 people of three different age groups with a questionnaire of 15 questions about the features of the UCFC application. The questionnaire contained questions about the usability of the prototype and the overall importance of certain concepts related to the effectiveness of UCFC application. Fig. 20 shows the survey results. The category being considered in Fig. 20(a) covers overall user friendliness, easy to use, ease of inputting information. From the graph, it is evident that a user friendly interface is the most important feature according to the users. The category shown in Fig. 20(b) reveals that the prototype requires enhancement in data representation, promptness, and ease of configuration issues.



(a) Ratings by users

(b) Ratings by users

Fig. 20. Usability study results of UCFC application.

9.4. Features of UCFC application

The carbon footprint calculator platform provides an important tool for individuals to assess their CO_2 emissions. It also allows application developers to develop such types of applications on top the UCFC platform. As this field continues to expand, accurate and transparent values will be needed to educate users and motivate effective responses on the individual and policy levels. The current proposal of UCFC provides some unique features such as:

- *Ubiquitous*: It is a mobile and ubiquitous application, i.e., it can be used at any time and any place if the user has a smartphone or PDA.
- *Adaptive*: It focuses on addressing a generalized solution to the carbon dioxide measurement problem. It also can be adapted to be used with other sensors too.

- *Sensor based*: Our proposed carbon footprint calculator is different from already existing calculators. Since this system can use the built-in sensors in the portable devices as a basic source of input. This sensor data will help the user keep track of carbon footprint with as minimum effort as possible.
- *User friendly*: It requires minimal end user interaction. Users of UCFC do not have to be technology experts.
- *Low cost*: It uses low cost and reusable sensors so that it can be cost effective. Moreover, it has a simple web interface, which can be used by the user to monitor the values of different sensors.
- *Ease of mobility*: It does not need Internet connectivity all the time. It needs WiFi/Internet connectivity only when it wants to upload data to the server. This allows it be used easily anytime and anywhere.
- *Customizable*: UCFC is designed in such a way so that it can be modified or customized by application developers.

10. Feature collection and design challenges of a fuel efficient mobile GPS application

The greatest contributor of CO_2 emissions in the average American household is personal transportation. Because transportation is inherently a mobile activity, mobile devices are well suited to sense and provide feedback about these activities. To demonstrate the use of our ubiquitous framework, we are also developing a mobile GPS application for the iPhone platform. Using a different computational model based on the regression analysis of the collected data, the application will be able to suggest the greenest/eco-friendly route to the user. Currently, we are working on the design and prototype implementation of such an application. In the following sections, we will discuss some important features and design challenges of such an application.

10.1. Features of a fuel efficient mobile GPS application

A mobile GPS application should provide most of the functionality available from existing GPS systems while equipping drivers with an easy way of planning routes that are environmentally aware. Some of the important features of such an application are:

- It suggests the greenest/eco-friendly route or the most fuel efficient route to the user.
- The application calculates a “fastest route” and “shortest route” in addition to the route with the least carbon emissions.
- It can calculate an individual's total carbon footprint over the course of an entire day from to his/her transportation.
- It is able to record carbon contributions and allows the user to compare his/her data to a pre-calculated average.
- It can suggest alternate forms of transit (like bus, train, etc.).
- It is able to track the user's travel pattern and can build a velocity profile figuring out how efficiently the user is driving.
- It is able to suggest the best transit routes that the user often uses. And it is also able to suggest that transit when the user is actually traveling under those circumstances.
- It allows the user to review their trips with details (e.g., what car was used or whether it was a public transit or a carpool).
- The application quantifies user's carbon footprint in terms of environmental impact and financial savings.
- It is able to warn the user about upcoming changes with sufficient time to react.

- The application allows individuals to post ride-share information.

This application is different from the Green GPS [13] system, since it is an application developed for a user's smartphone. Users do not need to carry a traditional GPS car navigation device. In the deployment of Green GPS, the authors use one such off the-shelf device for data collection purposes, called DashDyno [2]. The DashDyno's OBD-II scanner is interfaced to a Garmin eTrex Legend GPS [14] to get location data. Our application is entirely built on mobile devices where we collect the user's location information from the GPS sensor of the device.

10.2. Design challenges of a fuel efficient mobile GPS application

The application will consist of two major parts: the iPhone application and the framework. The basic challenge in developing such an application is to quantify the amount of carbon emissions that a person contributes to the environment through her driving activities. The computational model component of the OCFF framework needs adaptation for that purpose. But before discussing computational models and their corresponding parameters, we discuss some of the important design challenges of a mobile GPS application based on the iPhone platform:

- The iPhone-based application will allow a user to find the greenest route from one location to another. The application will also display statistics pertaining to an individual's carbon footprint over a given period of time. The mobile device application will accept a user's request for directions with start and endpoints as well as parameters for the algorithm (vehicle make, model, fuel type, etc.). Then it stores the vehicle profile information locally for future use. The application needs to submit the GPS location to the mapping API to obtain the correct map to display. The endpoints and algorithm parameters also needs to be submitted to the framework. The framework should resolve both the start point and the endpoint with the geocoding API (using both geocoding and reverse geocoding). This information along with the parameters can be used to calculate the routes. Routing information can then be displayed on the iPhone and the system can wait for a user selection.
- Once the user selects a route, the carbon footprint information from that route is committed to the database and the system should show the route on the map. The user also has the option of requesting a report for the system. The report will be based on a time period specified by the user (daily, weekly, monthly, etc.) and the application will request this information from the database. The framework will generate a report, which can be displayed on the iPhone. The iPhone's operating system, iOS, offers little choice when it comes to storing application data. Therefore, in order to sore any data on the application side, we use Apple's CoreData framework. The CoreData framework acts as a middleman between the application and a SQLite database.

Some other important design issues are:

- To send the routing parameters to the framework, the server and the application runs a programming language that supports performing network requests.
- To display the current location and overlay the routing information, the application must send the GPS location data to an external map API. The mapping API must be able to display the given location on the map.
- To resolve a user's location, we plan to rely on the use of an external geocoding API. The application will send the destination address to the geocoding API. The geocoding API will respond to the application's request with the GPS coordinates that correspond to the given address. The start point for routing can be determined using the iPhone's built-in GPS system. The application may also reference the geocoding API using GPS coordinates. The geocoding API takes the GPS coordinates and replies with an address.

- A key component in our application is the algorithm used to calculate the greenest, shortest, and fastest route from one point to another. The algorithm will use multiple inputs including vehicle type, gas type, time constraints, start point, and end point. The calculation will be done server side on the framework. It needs to use an appropriate language that provides a timely response.
- To display the user's selected route, the application must first receive the routing information from the framework API. The framework will return multiple routing options: shortest, greenest, and fastest. When the user chooses a route, the route statistics are stored in the database for modeling purposes. At the same time, the routing statistics are also stored locally for record keeping and displaying statistics. Finally, the application needs to overlay the routing information on the map provided by the external mapping API along with the route statistics. Part of quantifying a user's carbon footprint is being able to compare it to a representative average. Generic usage data will be stored in a database on the server to create an average CO_2 output profile. The server must be able to retrieve this information from the database and return it to the application for comparison.

10.3. Parameters of the computational model

The computational model of a mobile GPS application should be able to find out the most fuel efficient route for the user. The prediction model proposed in Green GPS [13] can be initially used for estimating the fuel consumption in a route for different car models. However, this model is not accurate enough as it does not consider real time traffic parameter in its model. Green GPS derives the model structure for fuel consumption from the basic principles of physics. They divide the parameters that affect fuel consumption into (i) static segment parameters, namely, numbers of stop signs, numbers of traffic lights, distance traveled and slope (ii) dynamic segment parameters, namely, average speed, and car specific parameters, namely, weight of the car and car frontal area. The derived approximated fuel consumption model is a function of the above parameters. The authors demonstrated that a single regression model is a bad fit for the collected data. Said differently, while a regression model that accurately predicts fuel consumption can be found for each car from data of that one car, the model found from the collective data pool of all cars is not a good predictor for any single vehicle. Hence, in a sparse data set (where data is not available for all cars), it is not trivial to generalize. However, in our computational model, we would like to collect data for more car model types and incorporate the real time traffic factor in the computation. We are currently working on the prototype implementation of such an application.

10.4. Benefits of a fuel efficient mobile GPS application

The main tangible benefit of a mobile GPS application is the ability of a user to quantify their carbon emissions based on driving activities. By providing the greenest route and comparing it to other alternate routes, the user can quantify the amount of CO_2 saved. By selecting the greenest route the user will also reduce the average amount of gas consumed and, in turn, save money. The most significant intangible benefit of this application is its potential to raise awareness about environmental sustainability. The term “green” suffers from a lack of concreteness. By being able to quantify carbon emissions and show users how their choices affect the environment, users can feel more empowered to make informed decisions that are environmentally and ecologically sustainable.

11. Conclusion and future works

Carbon footprint calculators are important tools for estimating CO_2 emissions and for providing information that can lead to behavioral and policy changes. However, current calculators produce estimates of carbon footprints that can vary by as much as several metric tons per annum per individual activity. Although most calculators are relatively new, their numbers are growing, and their CO_2 calculation methods are proliferating. Calculators can increase public awareness about CO_2 emissions and ways to reduce them. They can also affect the type and magnitude of emissions reduction efforts and offset purchases. However, a lack of dynamic and ubiquitous carbon footprint calculator has been observed lately. Therefore, in this paper we propose an Open Carbon Footprint Calculator application platform, OCFF that can determine a user's carbon footprint based on his/her dynamic activity. Given the prevalence and potential influence, OCFF can provide greater public benefit by providing greater consistency and ubiquity. Some of the important future research directions are as follows:

- •evelop and maintain a clouded knowledge base that will give developers access to a dynamic source of computational information that can be brought to bear on real-time sensor data.
- Develop necessary cloud-based web services as part of the OCFF that can perform some core calculations related to any type of carbon footprint application.
- Release and publish some cloud based web services that can perform some core calculations for various applications.
- To implement fully the fuel efficient mobile GPS application that can suggest the “greenest” route to the user.
- To develop a computational model using real-time sensor data for the application that can predict fuel consumption for various routes and for different car models.

References

[1] American Forests, Climate change calculator, http://www.americanforests.org/resources/ccc.

[2] Auterra, Dashdyno, http://www.auterraweb.com/dashdynoseries.html.

[3] Be Green, My carbon calculator, http://www.begreennow.com/index.php?utm_source=google&utm_medium=cpcutm_content=calculator&utm_campaign=carbon+footprint&gclid=CMnFy8S2-YgCFRJQVAodtk2Xpw.

[4] Bonneville Environmental Foundation, Carbon calculator, http://www.greentagsusa.org/GreenTags/calculator_intro.cfm.

[5] M. Buettner, B. Greenstein, A. Sample, J.R. Smith, D. Wetherall, *et al.* **Revisiting smart dust with RFID sensor networks.** *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNet ‘08),* ACM Press, New York, NY, USA (2008)

[6] Carbon Diem, http://www.carbondiem.com/product.html.

[7] Cognitive Walkthrough Strategy, http://www.pages.drexel.edu/∼zwz22/CognWalk.htm.

[8] Conservation Fund, Carbon zero calculator, http://www.conservationfund.org/?article=3128.

[9] D. Cook, S.K. Das, *et al.* **How smart are our environments? An updated look at the state of the art.** *Pervasive and Mobile Computing*, 2 (3) (2007), pp. 53-73

[10] A.K. Dey, D. Salber, G.D. Abowd, *et al.* **A context-based infrastructure for smart environments.** *International Workshop on Managing Interactions in Smart Environments* (MANSE ‘99) (1999)

[11] S. Easterbrook, T. Johns, *et al.* **Engineering the software for understanding climate change.** *Computing in Science and Engineering*, vol. 11, no. 6, IEEE Computer Society Press, CA, USA (2009) pp. 65–74

[12] EPA, EPA's personal greenhouse gas calculator, http://yosemite.epa.gov/oar/globalwarming.nsf/content/ResourceCenterToolsGHGCalculator.

[13] R.K. Ganti, N. Pham, H. Ahmadi, S. Nangia, T. Abdelzaher, *et al.* **Green GPS: a participatory sensing fuel-efficient maps application.** *Proceeding of the Conference on Mobile Systems, Applications, and Services*, ACM Press (2010) pp. 151–164

[14] Garmin eTrex Legend, www.8.garmin.com/products/etrexlegend.

[15] GooglePowerMeter, www.google.org/powermeter/howitworks.html.

[16] Interactive United States Energy Use Comparisons, http://www.eredux.com/states/.

[17] L. Liu, H. Zhang, S.I. Ahamed, *et al.* **Some thoughts on climate change and software engineering research.** WSCRCC (2009)

[18] J. Padgett, S. Paul, C. Anne, J.H. Clarke, M.P. Vandenbergh, *et al.* **A comparison of carbon calculators.** *Environmental Impact Assessment* (2008) pp. 106–115

[19] SafeClimate, Calculator, http://www.safeclimate.net/calculator/.

[20] A.P. Sample, D.J. Yeager, P.S. Powledge, J.R. Smith, *et al.* **Design of a passively-powered, programmable platform for UHF RFID systems.** *Proceeding Conference on RFID '07* (2007), pp. 149-156

[21] D. Sarah. **The Effectiveness of Feedback on Energy Consumption, Technical Report.** Environmental Change Institute, University of Oxford (2006)

[22] B. Shui, H. Dowlatabadi, *et al.* **Consumer lifestyle approach to US energy use and the related CO_2 emissions.** *Energy Policy* (2005), pp. 197-208

[23] TerraPass, For flight, http://www.terrapass.com/flight/index.html.

[24] TerraPass, For home, http://www.terrapass.com/home/index.php.

[25] TerraPass, http://www.terrapass.com/road/index.html.

[26] B.P. Weidema, M. Thrane, P. Christensen, J. Schmidt, S. Lokke, *et al.* **Carbon footprint: a catalyst for life cycle assessment.** *Journal of Industry Ecology*, 12 (1) (2008), pp. 3-6