February 2008

On the Feasibility of an Energy Reporting System in London - City Knowledge, the Merton Rule, and Building Birth Certificates

Benjamin Wray Lessard
Worcester Polytechnic Institute

Brian M. O'Keefe
Worcester Polytechnic Institute

Christopher D. Scalabrini
Worcester Polytechnic Institute

Robert F. Bernard
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/iqp-all

On the Feasibility of an Energy Reporting System in London

City Knowledge, the Merton Rule, and Building Birth Certificates

An Interactive Qualifying Project Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
by

_____

Robert F. Bernard

_____

Benjamin W. Lessard

_____

Brian M. O'Keefe

_____

Christopher D. Scalabrini

Date: February 21, 2008

Approved:

_____

Professor Malcolm Ray, Advisor

_____

Professor Constance Clark, Advisor

Abstract

The Borough of Merton has decided to look into City Knowledge as a tool for better monitoring of the Merton Rule. To obtain the initial information for this rule the Merton Council requires developers to submit Building Birth Certificates indicating what carbon reduction methods they will implement. These forms, however, are cumbersome for the developers to complete and time consuming for the Merton Council to evaluate. The goals of this project are to use the principles of City Knowledge to lay the groundwork for the prototype of the monitoring system as well as to create an intuitive online entry form to replace the current Building Birth Certificates.

Executive Summary

The London Borough of Merton, to combat rising carbon emissions, implemented a policy known as the Merton Rule. The Merton Rule requires that all new developments reduce their carbon emissions by ten percent by using onsite renewable energy sources. Previously, developers met this requirement via e-mail with the Principal Environment Officer, Adrian Hewitt. The result was that the final agreement was stored in Adrian Hewitt's mailbox, making it difficult to retrieve later. In addition, this complicated monitoring of ongoing compliance with the Merton Rule.

The Borough of Merton enlisted the help of WPI to resolve these problems. This project was the first of three in the series. The first project was to develop a web-based system for gathering the renewable energy source data and to provide a framework for real-time updates on a development's renewable energy use. The second project is responsible for research into the various monitoring devices available and the selection of one of these for use. The final project is to link the first two projects and to create a presentation package for Adrian Hewitt's presentations to the Greater London Authority.

The first objective for Team One was to develop a web-based entry form for developers. Merton would use these web pages to gather from the developers the information that they formerly exchanged via e-mail. The form would compile the data and store it in a database for easy retrieval in the future. The Team's second key objective was to develop a prototype system for monitoring the renewable energy used in Merton. This would enable Merton to enforce compliance with its Merton Rule. The reporting program would handle collection of data from the sensors, relay of the data to Merton, and comparison with the baselines set via the web forms. The system would also be responsible for relaying data from Merton to the Greater London Authority, as they have requested renewable energy reports from each of the Boroughs.

Upon arriving in Merton, the Team consulted Adrian Hewitt to determine the information he required on the entry form. He was able to provide the Team with the Excel spreadsheet he used to store the development details, which he called a "Building Birth Certificate," as well as a calculator for the renewable energy requirement. After some review, the Team revised this "Building Birth Certificate" into the prototype for RELIEF, which was to be the new entry form.

In parallel with the design of RELIEF, the Team created the groundwork for the reporting system. Four layers compose the system: a base framework, a reporting back-end, a front-end, and a web interface. The base framework contains the generalized features of networking, security, database access, and plug-ins. The back-end utilizes the framework for the more specific needs of the reporting system. The front end allows users to interface with the system. Finally, the web interface eliminates the need for the program's installation on all computers used for viewing.

During the course of the project, the Team developed guidelines for the two subsequent projects. With these guidelines, the Team hopes to ease transition between the projects and to reduce the time to resume the project. Some of the guidelines produced include a developer's guide to the interface between the system and sensors and a guide to the security features.

Table of Contents

Table of Figures

Introduction

Global warming is the theory that the overabundance of greenhouse gases in the atmosphere is causing the average temperature of the earth to increase gradually. These greenhouse gases allow the majority of solar energy to pass through; after the earth's surface re-radiates the ultraviolet radiation as infrared radiation, these gases absorb and re-emit a portion of the radiation back down to the earth.[1] As these gases accumulate, the re-emission of more radiation down to the surface causes the average temperature of the earth to rise. A United Nations study (2006) confirmed that people worldwide were releasing greenhouse gases into the atmosphere faster than the gases were being recycled. With the release of 7.91 billion metric tons of carbon in 2004[2], reducing the amount of carbon dioxide released into the atmosphere has become the focus of many governments worldwide.

An international effort to battle this growing problem came in the form of the Kyoto Protocol in 1997. The United Nations created the treaty as a worldwide attempt to slow the process of global warming. This treaty created different standards for each nation that allow all the countries to reduce carbon emissions based on their output and their available resources to reduce emissions effectively. In 2007, the U.N. held a conference in Bali to further these global initiatives started by the Kyoto Protocol. This conference reviewed national reports of greenhouse gases to decide further action. One major occurrence at the conference was the ratification of the Kyoto Protocol by Australia, leaving the United States as the only country left to ratify the treaty.

The British Government has taken a leading role in the effort to identify innovative ways to reduce the emissions of greenhouse gases, and is hoping to reduce the country's overall emissions by 60% by 2080. As part of this effort, the government has established "Energy Actions Areas" in four London boroughs (Merton, New Wembley, Southwark, and Barking). Each "Energy Action Area" is testing the feasibility and effectiveness of different innovative technological and policy approaches in reducing greenhouse gas emissions. One of the areas

---

[1] (Energy Information Administration, 2007)

[2] (Marland, Boden, & Andres, 2007)

that made the largest contribution to meeting these goals is the Borough of Merton with its Merton Rule.

Merton is leading the way for much of the United Kingdom with the development of the eponymous Merton Rule. The Merton Rule states that buildings must reduce their carbon emissions by 10% through the implementation of onsite renewable energy resources. Many other boroughs, regions, and cities have adopted the Merton Rule including the boroughs of Yorkshire, Manchester, and Greenwich[3]. Combined Heat and Power generators (CHP Generators), bio-fuels, and wind turbines are all methods currently employed in new buildings in the effort to reduce emissions; however, several problems still exist. Within Merton, there is no current method of tracking the amount of renewable energy employed by the buildings. In addition, there is no effective way to report findings to the national government.

The Council of Merton wants to solve the aforementioned problems using City Knowledge. City Knowledge is a framework designed by Dr. Fabio Carrera to improve methods for gathering, storing, and accessing data about a city. Specifically, Merton would like to connect its existing Geographic Information System with new information on compliance with the Merton Rule. The solution proposed requires each new building to have a "birth certificate" indicating the measures it is taking toward compliance. The current GIS could then incorporate Building Birth Certificate data. Through the cooperation of these systems, members of the Council will have instant access to a map of Merton with compliance indicators in place. Merton has requested the assistance of WPI to help solve these problems

This project will be divided into three separate parts to be completed by three Teams from Worcester Polytechnic Institute. The First Team will investigate the use of Building Birth Certificates to predict Merton Rule compliance. Based on the data Merton currently has and the additional data the Borough wants, the Team will implement a web-based system to enable developers to complete Building Birth Certificates and submit them electronically. The second key goal for Team One is to delve into the software systems to automate the energy reporting process. Eliminating the need for manual reporting will streamline the compliance calculations, enabling the Council to collect data directly from buildings, while simultaneously providing the same data to the appropriate central government agencies. Various departments within the

---

[3] (The Council of Merton, 2007)

Council of Merton must also have access to the Merton Rule compliance data. For example, the Planning and Projects Department within the Merton Council could use the data to find trends to improve future projects.

Background

The issues of global warming and greenhouse gas emissions only recently have begun receiving the appropriate attention. Many countries are now taking action to reduce greenhouse emissions, particularly carbon emissions. At the time of writing, 174 countries have ratified the Kyoto Protocol. As one of the countries to sign and ratify this, the United Kingdom hopes to take the lead in carbon reduction. The U.K. government designated four boroughs as Energy Action Areas, and as one of these, the Borough of Merton has implemented the "Merton Rule." To monitor compliance with this standard, the borough plans to use an already-implemented system known as City Knowledge, which will compile into a single database relevant data on energy consumption and emissions in various sectors and activities in the borough.

Climate Change

Greenhouse effect

Without the greenhouse effect, the average temperature of the earth would plummet from the current comfortable 14 degrees Celsius to a frigid -18 degrees Celsius.[4] As shown in Figure A, only a portion of this solar energy penetrates the atmosphere and hits the surface. The surface then re-radiates the energy back into the atmosphere. The greenhouse gases there absorb and re-emit the radiation back into the lower atmosphere and surface of the earth heating the surface.

---

[4] (National Climatic Data Center, 2007)
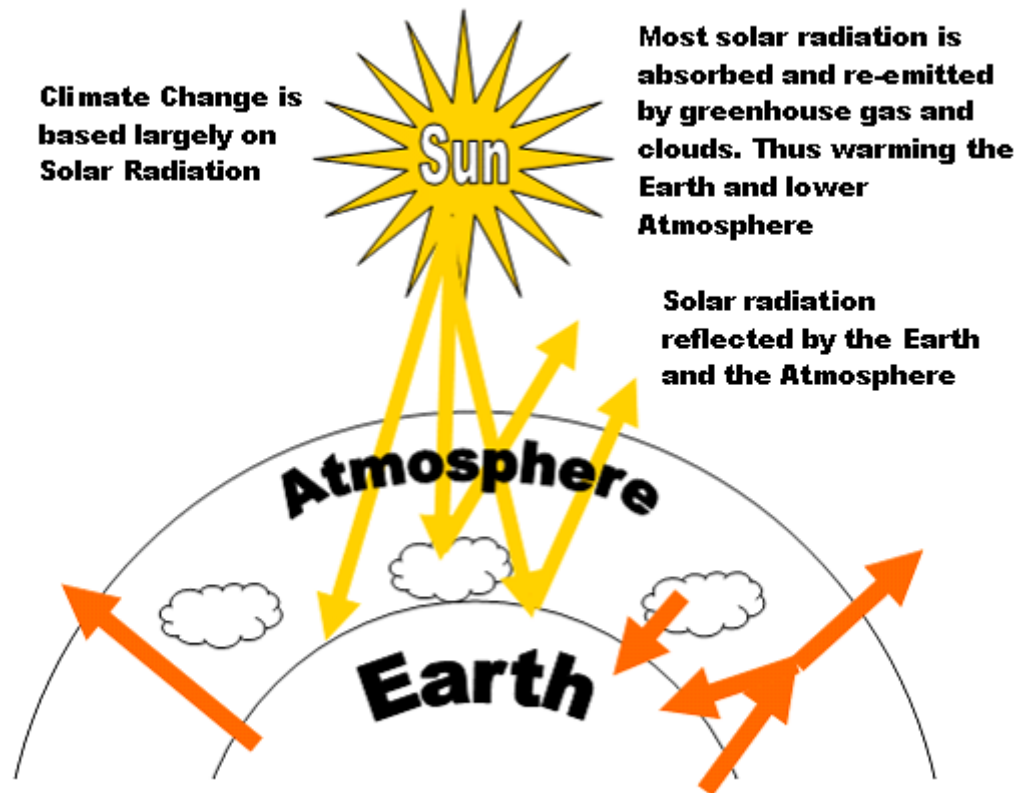
Figure A: The Greenhouse Effect[5]

Brief History

In the early 1900's, the Swedish chemist Svante Arrhenius and the American geologist Thomas Chamberlain hypothesized that the amount of carbon dioxide in the atmosphere was directly influencing the average temperature of the planet.[6] However, this sank into the background of world politics with the outbreak of World War I. It resurfaced in 1938 during a study by the English engineer Guy Stewart Callendar where he suggested that a doubling of the CO_2 in the atmosphere could increase the global average temperature by two degrees Celsius.[7] The scientific community, though, dismissed this as having insufficient supporting data. Although further research was conducted, no universally accepted conclusion was accomplished until 2007 when the Intergovernmental Panel on Climate Change (IPCC)

---

[5] (Le Treut, et al., 2007)

[6] (Weart, 2007)

[7] (Le Treut, et al., 2007)

established that human-induced greenhouse gases were undeniably affecting the global climate.[8]

Although there are other greenhouse gases (e.g., Sulfur hexafluoride (SF_6), Hydrofluorocarbons (HFCs), Perfluorocarbons (PFCs), nitrous oxide (N_2O), and methane (CH_4)) carbon dioxide is the most detrimental. A study done by the UNFCCC (United Nations Framework Convention on Climate Change), in preparation for the convention in Bali from December 3-11, 2007, found that the amount of CO_2 released into the atmosphere by Annex I countries (i.e., developed countries like the United Kingdom, the United States, Russia, etc.) greatly exceeds the release of other greenhouse gases. Figure B shows the contrast between years 1990 and 2005 for comparison.
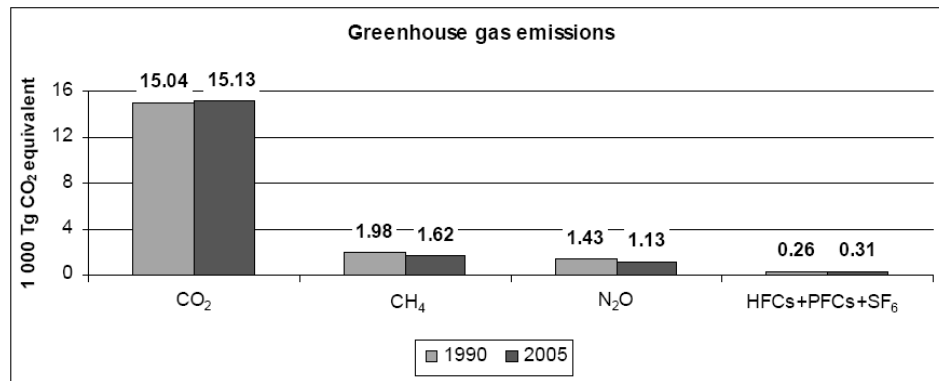
Figure B: Annex One Party Greenhouse Gas Emissions[9]

---

[8] (United Nations Framework Convention on Climate Change, 2005)

[9] (United Nations Framework Convention on Climate Change, 2007)
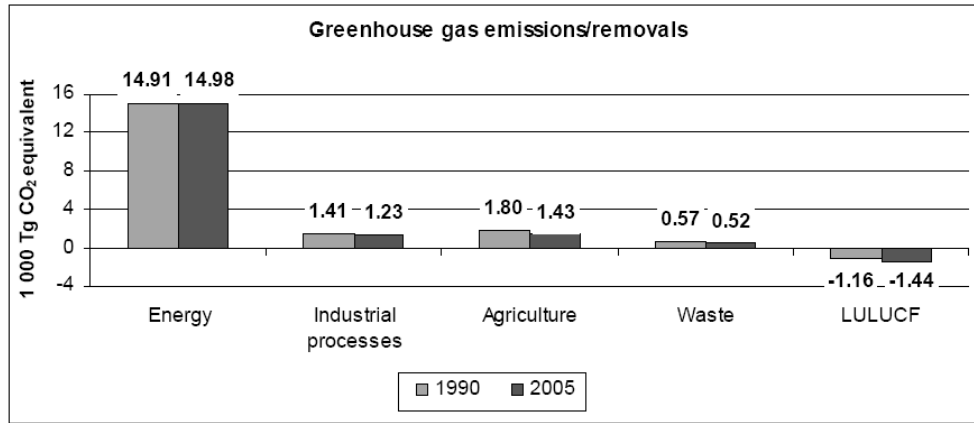
Figure C: Annex One Party Greenhouse Gas Emissions/Removals by Sector[10]
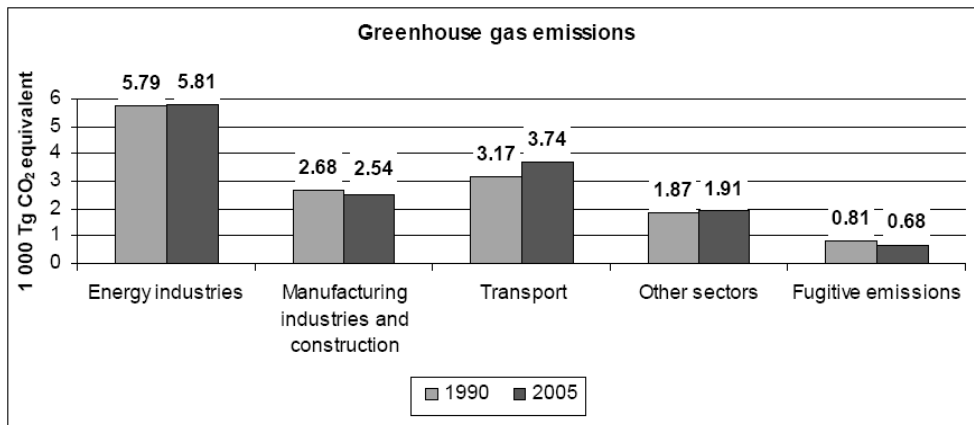
Figure D: Annex One Party Greenhouse Gas Emissions in the Energy Sector[11]

As shown in Figure C, this study also divided the major CO_2 contributors by sector. The energy section dominates the others because this includes the burning of fossil fuels to energize businesses, homes, and transportation. To explore the largest contributor the study further divided the Energy sector in to sectors of is own. As shown in Figure D, the largest contributor here is the energy industries, including power plants. While it is noted that transport is the second highest producer of CO_2, the nature of the project precludes this from the current study.

---

[10] (United Nations Framework Convention on Climate Change, 2007)

[11] Ibid.

Nature and extent of climate change

As shown in Figure E, the average global temperature, based on the average global temperature between 1961 and 1990, has been consistently increasing since 1856.
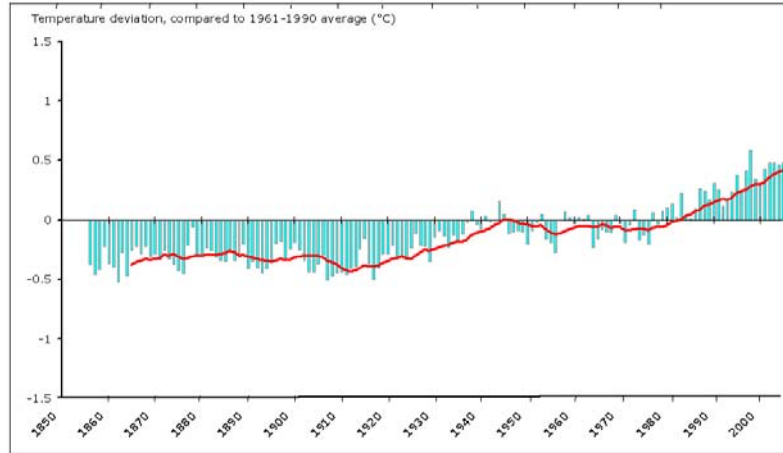
Figure E: Temperature Deviation[12]

The IPCC predicted that, assuming no change occurred in policies concerning climate change, the global average temperature would rise by 1.4 to 5.8 degrees Celsius between the years 1990 and 2100.[13]

Nature and extent of impacts

This increase in temperature would have drastic effects upon the entire planet. One of the largest concerns is that a rise in the average global temperature would both increase the melting rate of the globe's icecaps and cause greater thermal expansion of the oceans. The IPCC estimated that between 1990 and 2100 the sea level would rise by 0.292 to 2.883 feet[14]. This would cause major flooding in many coastal areas.

A rise in temperature would also affect weather patterns. A warmer global average temperature would mean quicker evaporation. This increase would affect the precipitation levels

---

[12] (European Environment Agency, 2006)

[13] (European Environment Agency, 2005)

[14] (Michaels, Balling, & Davis, 2005)

causing rainstorms to become more frequent and more violent. Wildfires would also occur more regularly with the faster evaporation of moisture causing drier forests more susceptible to fire.[15]

International Initiatives

United Nations Framework Convention on Climate Change

Before the Kyoto Protocol, the United Nations created the Framework Convention on Climate Change, also known as the Rio Declaration, to attempt to control the growing problem of greenhouse gas emission, in particular carbon dioxide. The convention was initially open for signature on May 9, 1992 and fully ratified on March 21, 1994. One hundred and ninety one countries ratified the convention, unlike the Kyoto Protocol. This is because the convention, also unlike the Kyoto Protocol, does not set limits on greenhouse gas emissions and does not include any sanctions for failure to adhere to its rules. What it did, which the Kyoto Protocol did not do until five years later, was to recognize that greenhouse gas emission is an internationally shared problem and only through international cooperation can a solution occur. This Convention and Agenda 21 facilitated the creation of the Kyoto Protocol within the decade.

Agenda 21

In 1992, local authorities in Europe came together to develop an environmental action plan in response to the Rio Declaration announced by the United Nations earlier that year. The Rio Declaration intended to raise awareness about environmental issues, especially climate change and biodiversity, and to encourage countries to explore ways to address these problems.[16]

Kyoto Protocol

The main objective of the protocol is to attain "stabilization of greenhouse gas concentrations in the atmosphere at a level that would prevent dangerous anthropogenic interference with the climate system."[17] In signing the protocol, a country agrees to reduce its greenhouse gas emission by a certain percentage by the year 2010. Depending on each country's emission levels recorded in 1990, each signatory is required to meet specific goals.

---

[15] (Natural Resources Defense Council, 2007)

[16] (UN Department of Economic Affairs, 2005)

[17] (United Nations, 1992)

For example, the United Kingdom is obliged to reduce its emission levels by eight percent by 2010.
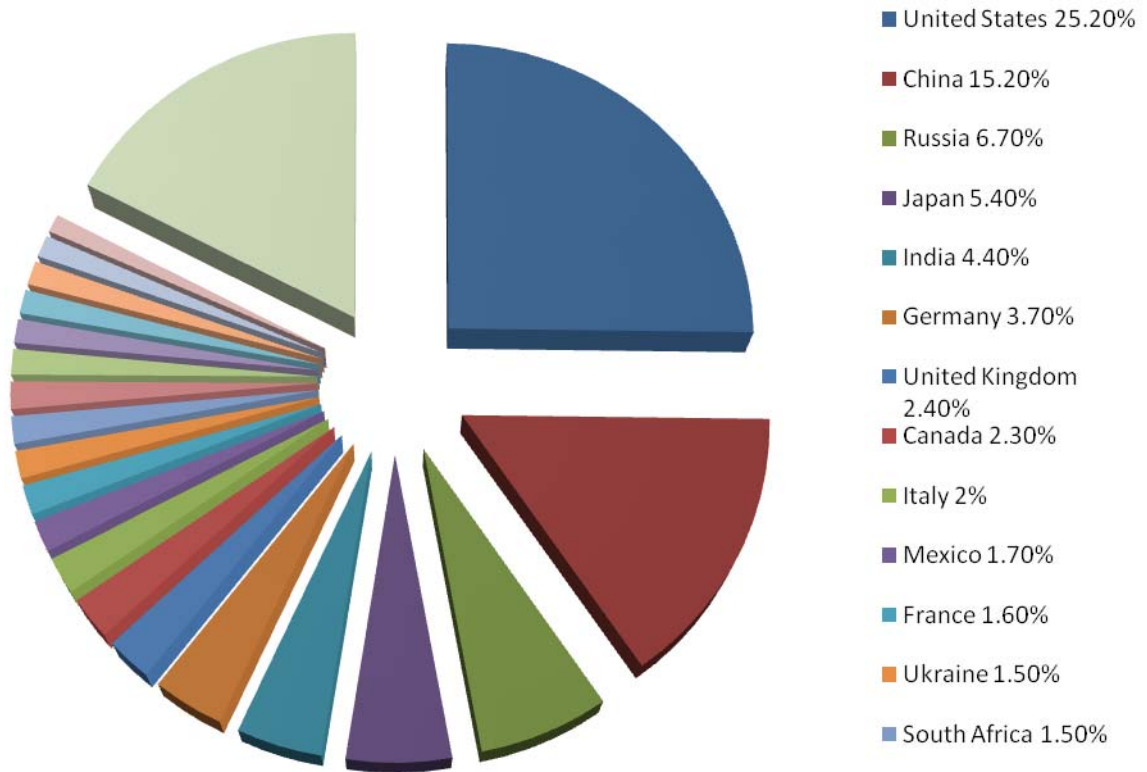
Figure F: CO$_2$ Emissions by Country[18]

When a country signs the Kyoto Protocol, it has two options. It can sign the treaty, but then only report and monitor its emissions, or it can sign and ratify, which requires active reduction of its emissions. As shown in Figure F, the United States is currently the only country in the world that has signed but not ratified the treaty.

---

[18] (World Resources Institute, 2003)
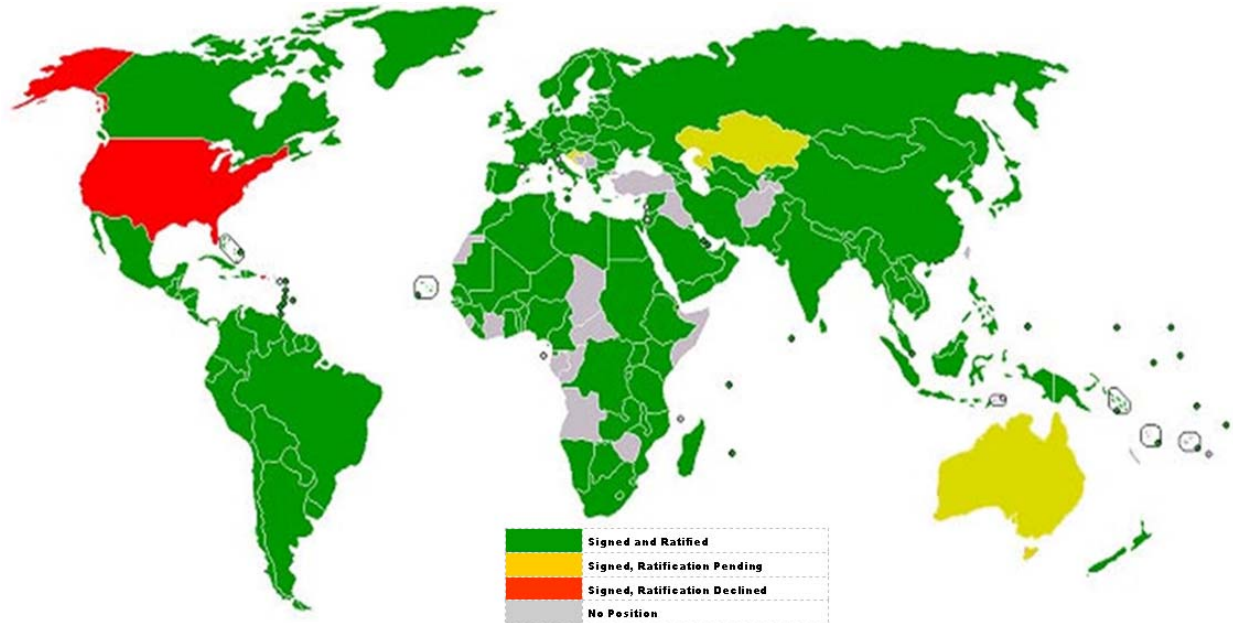
Figure G: Countries Signing the Kyoto Protocol[19]

United Kingdom's Involvement

The United Kingdom has pledged not only to reduce its emissions by the required eight percent but also to surpass that by requiring a twenty percent reduction by 2010 and a sixty percent reduction by 2080.[20] The United Kingdom created several 'Energy Action Areas' to help achieve the self imposed goals. These areas are responsible for implementing programs that will help, through different methods, obtain the national goal for the Kyoto Protocol. Some 'Energy Action Areas' have started to create programs that will aid in the collection of information pertaining to the Kyoto Protocol.

London's "Energy Action Areas"

The Greater London Government designated four areas as "Energy Action Areas" to help lead the effort in developing environmentally friendly communities. The areas are "a way of showcasing work that takes forward the objectives of the energy strategy."[21] Each different area focuses on a particular method of energy reduction. Barking is working with a new development

---

[19] (Fullerton College, 2007)

[20] (DeMello, Guzman, Luk, & Robert, 2006)

[21] (London Energy Partnership, 2005)

that will include 4000 homes designed to reduce the carbon effect of the homes and areas surrounding them. New Wembley is an area that is developing "3,700 homes as well as hotel, leisure, retail office, and community space." The company developing this area is dedicated to the energy action program thus creating a foundation for continued renewable energy growth. The Borough of Southwark is focusing on reducing carbon emissions already produced in the town. The local government plans to cut emissions by 75 percent by employing the use of bio-fuels, wind turbines and combined heat and power (CHP).[22] As mentioned above, London has created new programs of reducing carbon emissions that will allow the city to reach its end goal of 60 percent reduction by 2080.

Renewable Energy Sources

There are two different approaches to using renewable energy sources: there are heat generating renewables and electricity generating renewables. The pro and cons of each, as well as a few examples, follow.[23]

Heat Generating Renewables

The focus of these renewable sources is to pre-heat/cool the water needed for building heating/cooling systems and for personal use. Since these systems reduce the energy required to heat/cool the water, the boiler will need to produce less heat, thereby reducing its carbon emission. The major downside of these systems is the time it takes before the water is ready for use. The system also becomes useless if the building does not use the water.

Solar Hot Water Systems

The basic principle for solar heating systems is that a small pump drives water to a series of tubes located on the roof that are sealed in a bigger vacuum tube. The vacuum allows no heat loss while the water absorbs radiation from sunlight. The water, now preheated, runs to a tank that holds the water until needed by the user. These systems cost £2,850 - £3,500 per square meter.

Ground Source Heating & Cooling

Ground source heating and cooling is essentially the same principle as the solar hot water system but instead of solar energy; it uses the energy stored in the ground. The system

---

[22] (DeMello, Guzman, Luk, & Robert, 2006, pp. iii-vii)

[23] (Maunsell, 2004, pp. 21 - 88)

pumps water through a series of tubes buried underground because it is about 5-10 ºC warmer during the winter months. During the summer, when the temperature is 5-10 ºC colder underground, some companies have used the system to pre-chill the air for the cooling cabinets. The cost of the system is relatively low, based on the technology used. The average additional cost is about £2,500 for a residential home and about £50,000 for 50% of the heating demand of a 3000 m² office building.

Biomass Heating and Biomass Combined Heating & Power (CHP)

Biomass systems work by burning wood pellets to produce heat. The CHP adds a water tank over the exhaust to heat the water. The fuels for biomass are mostly small wood pellets, delivered by truck. Companies consider this a renewable energy source because it uses the energy from burning wood. While fossil fuels are finite, trees are easily replaceable. The major problem with this system is that it still produces carbon and may produce other harmful gases.

Electricity Generating Renewables

The purpose of these renewable sources is to produce additional electricity through environmentally friendly methods. Electricity gained through these methods reduces the electricity needed from carbon producing power plants, thereby reducing the carbon the plants release. One of the disadvantages of these systems is their dependence upon outside factors. Wind turbines need wind and solar panels need daylight. The greatest advantage of energy generating renewables is that the electricity never goes to waste. Even if the site generating it does not use the electricity, the site can resell the excess to the grid for use by another institution.

Wind

Wind turbines are among the most cost effective methods of renewable energy generation currently available. Quiet wind turbines are the most popular in small urban applications. However, public opinion is a large problem with these wind turbines because of the size and possible noise. The basic construction is a horizontal axis with three rotating blades that drive a generator that produces the energy. The average cost of a wind turbine depends on the amount of energy needed. Currently the initial cost range is £2,500 - £5,000 per kW produced. Once installed the only cost would be for any necessary maintenance or replacement parts.

Photovoltaic

Photovoltaic systems (also known as solar panels) currently have the best cost-to-benefit ratio of the renewable energy generation methods. The panels consist of a semi-conducting material that reacts with light to produce electricity. A common misconception is that the panels need direct sunlight to operate. Although the panels produce more energy in direct sunlight, general daylight will still produce a reasonable amount of energy. Average cost per kilowatt is about £5,000 - £8,000. Once installed there is no additional cost. Another benefit to the photovoltaic systems is that they can be installed anywhere there is sunlight. Roofs, windows, and siding are viable options for the systems.

Merton Rule

Merton

The Borough of Merton has developed the 10% Rule as a method to reduce the carbon emissions of buildings in the borough. Merton has geared the current plan toward new buildings in Merton; however, there are plans to implement this system in every municipal building in the Borough. The target developments are generally those with more than 10 residential units or non-residential buildings with over 1000 m^2 of floor space. The basic principle of the Merton rule is to reduce the carbon emissions by 10% using onsite renewable energy resources like solar water heaters, small wind turbines, or even combined heat and power generators (CHP).[24] As the push for reduction in emissions has increased, so has the expansion of the Merton Rule to other Boroughs.

Other Boroughs

Twenty-three different governing bodies in the U.K. have adopted the Merton Rule as the primary method of reducing emissions, and 142 are currently in the process of full adoption.[25] Since the Merton Rule is relatively easy for many different boroughs and other forms of government to implement, it has gained widespread acceptance.

---

[24] (Environmental Protection Agency, 2007)

[25] (The Council of Merton, 2007)

1APP System

The Team researched what information building applications contain in Merton and how the developers submit them to the Development Control Department. Applications are currently submitted using paper applications. The United Kingdom, however, is in the process of adopting an online applications system named 1APP to increase application efficiency. This system will provide a series of forms, dependent on the type of development, to be completed online. More forms will need to be completed per building than under the current system, but they will also be more specific and less time consuming. These applications will then be submitted by developers via the Internet and reviewed by Development Control departments in their respective Boroughs. After April 6, 2008, 1APP will become the only legal way to apply for a building permit in England or Wales.

Building Birth Certificates

The Borough of Merton's Principle Environment Officer, Adrian Hewitt, has proposed the concept of "Building Birth Certificates" for monitoring compliance with the Merton Rule. Similar to a human birth certificate, Building Birth Certificates will describe aspects of the building's design as specified when the Council approved the building plan. They will provide a baseline for initial Merton Rule compliance measurements and support tracking of progress toward compliance.

The Development Control Department in Merton holds information relevant to the buildings in the new 1APP system. Information such as the address, type of building, the developer, and development control officer are included in this system. More specialized fields for renewable energy installations, including window area, surrounding land, and predicted heat and energy usage, are not yet included in the database. Creating an additional database for the renewable energy information will allow all information about a building to be accessible, ensuring each building has complied with the 10% Rule.

City Knowledge

Overview

City Knowledge, in the sense proposed by Doctor Fabio Carrera, is not a collection of data, but rather a set of principles used in conducting assessments. In his dissertation, Carrera describes six such principles: the "middle-out approach," distinct informational jurisdictions, atomic data organization, a sustainable update mechanism, sharing of information, and inter-

agency coordination.[26] For a City Knowledge system to succeed, Carrera suggests, the "middle-out approach" is fundamental; coordination of both a bottom-up development driven by the data collectors, and a top-down approach directed by high-level management, are vital to the sustainability of such a system. Carrera also warns against a single jurisdiction over the collected data, instead suggesting that a data collection agency should maintain its own data with oversight from high-level managers. In addition to the distributed data collection, City Knowledge should use an atomic data structure distributing the data storage across departments such that each stores the data it collects. A City Knowledge implementation would be useless without updated data and a mechanism for sustained updates; Carrera suggests several methods for ensuring the availability of updates.[27] City Knowledge also suggests that data be made available to others voluntarily; that is, information does not need to be shared unless a mandate exists requiring the data be shared, or the data's owner desires it to be shared. Finally, Carrera notes the necessity of proper communication between data collection agencies on standards of data format.[28]

Once a municipality applies the City Knowledge principles, it must provide a way to store, view, and change the data it collects. When Carrera established City Lab, he began work on a free, open-source client for GIS access, known as LOUIS (the Local Online Urban Information System), based on the MassGIS OLIVER client.[29] In a future version, he intends to support the OpenGIS standard. Once complete, LOUIS will provide web-based access to an "urban information system targeted primarily to municipalities and other local authorities."[30] Thus far, small towns have supported LOUIS's development through small grants. LOUIS has also received support from several international sites, including the cities of Venice, Italy and London, United Kingdom.[31]

---

[26] (Carrera, City Lab @ WPI, 2006)

[27] See (Carrera, City Knowledge, 2004, p. 210) for examples

[28] (Carrera, City Lab @ WPI, 2006)

[29] Ibid.

[30] Ibid.

[31] Ibid.

City Knowledge and the Merton Rule

City Knowledge and its governing principles are easily applicable to the Merton Rule. In accordance with the middle-out approach, the Council of Merton will provide oversight for collection of energy use data but will not directly collect the data. The building planner will be responsible for providing the data he has on the building's design. In addition, the Council of Merton has requested that the compliance data be shared with the government of the United Kingdom. Sustained updates may cause more work for members of the Council but will enable the Council to enforce continued compliance with the Merton Rule.

The Borough of Merton already employs a GIS; by using City Knowledge's principles of atomic data organization and distinct informational jurisdictions, the Projects and Planning Department will maintain responsibility for the data collection with regard to Merton Rule compliance. They will also need to examine the existing GIS data and determine the best option for integrating compliance data to facilitate sharing. The use of a GIS would facilitate the creation of a map of Merton with overlaid compliance markers.

Methodology

The end goal for the Borough of Merton is to have a fully functional prototype of a data collection system based heavily upon City Knowledge principles by the end the summer 2008. This prototype will be a proof-of-concept for a future government-funded project. Work started in January 2008 with the design and implementation of a web-based data collection system to fill out building birth certificates pertinent to onsite renewable energy resources. The First Team (i.e. January and February) created the initial coding to establish the foundation of the energy reporting system. The second stage will occur in March and April of 2008 with extensive research and installation of sensors in buildings to monitor and report energy usage for compliance with the Merton Rule. Finally, the third stage is to complete the energy reporting system by producing appealing user interfaces for the online entry form and energy reporting systems. This Third Team will also test the systems to ensure universal usability and develop a protocol for the expansion of these systems outside Merton.

The First Team's project has two primary goals: (1) explore the use of Building Birth Certificates to monitor compliance with the Merton Rule and create a preliminary online data entry form; and, (2) create a proposal for a national system to report energy usage data to the UK government. Listed below are the tasks for the First Team in summarized form. They are described in further detail later.

- Building Birth Certificates for Predicting Merton Rule Compliance
    o Gather energy usage data and carbon-reduction method data from the four Energy Action Areas, as well as two other boroughs that have implemented the Merton Rule.
    o Analyze collected data to infer levels of success based on already implemented methods for energy and carbon emission reduction. The Team will measure success based on several factors, including financial viability, emission reduction, and ease of maintenance.
    o Expand upon current Building Birth Certificate structure to reflect results from the data analysis.
    o Perform Needs Study to determine users for Building Birth Certificate Entry Form.

- o Design an online Building Birth Certificate entry form. The Needs Study provides the content for the form.
- o Perform a Usability Study on the Building Birth Certificate entry form to determine whether it meets the needs of users.
- o Perform Pilot Test on Building Birth Certificate entry form in order to locate errors in the entry form.
- Proposal for Energy Reporting at the National Level
  - o Design a flexible and usable code framework for communication and sensor updating.
  - o Design prototype program for collecting and reporting energy data.
  - o Design specification for computer that will be a data server.
  - o Perform minor tests on the system in real-world situations.
  - o Provide code base for a web-based energy reporting system.

Building Birth Certificates

Although building birth certificates currently include information regarding renewable energy, there is no database specified for storing this information. Creating an entry form and database for this information will be the main task during the First Team's project. This will allow building contractors and developers to input the information directly into the database. The fields for data entry will come from a complete revision of the current Building Birth Certificate form and information from current buildings that have implemented renewable energy resources.

Energy Use Data Collection

The Team's main tasks for the research into Merton and other boroughs are as follows: research into the data needed by Merton, draft a preliminary revised building birth certificate, and have other boroughs review it for relevancy and usability. Data collection methods will be based on liaison's suggestions and further research done while in Merton.

- Conduct research pertaining to renewable energy sources.
  - o Determine approved methods for Merton Rule compliance.
- Construct preliminary revised Building Birth Certificate.
  - o Include fields suggested by Merton.
- Discuss with the liaison possible contacts outside of Merton.
  - o Which Merton Rule boroughs should be contacted?

- Revise Building Birth Certificate further according to the responses from other boroughs.

Building Birth Certificate Expansion

The Building Birth Certificate proposal, as described previously, represents information about new buildings. The actual structure of a Certificate is a simple table in Microsoft Excel with fields for each type of information. Currently, Adrian Hewitt populates the form. The Team's goal is to create a system where the property developers populate the fields on-line and, once completed, the Planning and Projects Department will have access to the information. The entry form will contain information about both planning and environmental impact, which will be placed into the proposed City Knowledge data storage system, allowing the information to be shared among departments but not compromising format or the actual data.

Online Entry Form Needs Study

The purpose of the Needs Study is to pinpoint exactly what the users of the Building Birth Certificate entry system need and want. In order to perform a relevant and accurate needs study, the project team must locate appropriate participants. The participants that will be contacted are contractors, building planners, and individuals in the Planning and Projects Department. The project team will then conduct interviews with each participant, either via telephone or email. If the project team interviews the participant via telephone, the Team will employ the use of either Skype or a speakerphone in order to facilitate a second person to fulfill the roll of documenter. The project team will contact as many individuals as possible within the time restrictions of seven weeks. The interview will cover topics such as what the individual feels is important with the Building Birth Certificate process, as well as any possible concerns about the implications of using an internet-based system in lieu of a paper-based system. A short preamble will precede each interview describing its purpose, in addition to a short survey to collect demographic information. A sample interview record-sheet is in Appendix C.

Online Entry Form Construction

Online entry forms provide an accessible way for individuals to provide information to organizations that request it. Creating the foundation of an online entry system for Building Birth Certificates is another objective of the project team. The Needs Study above will provide much of the required information for the data that must be included. Depending on the type of server and operating system the borough uses, the choices of programming and scripting languages

will vary. In order to obtain such information, the project team will need to contact the Information Technology professional on staff at the Borough, either in person or by telephone.

Online Entry Form Usability Study

The group will design and run a usability study based on the website draft, in which participants will test the various features. The group will observe the users at work performing a set of predetermined tasks; the tasks will be heavily based on the information collected during the needs study. The users will be selected from the same pool as the participants for the needs study. This selection process will ensure that the users selected are appropriate to the purpose of the form. In order to do this, a computer will be required. In addition, the study will require either the project team goes to the individual, or the individual will need to go to the study site. Another possibility is a sort of web tracking system that will monitor the user's actions. Each user would require approximately a half hour of time, whereas a simple interview would require less. In addition, the project team likely will interview individuals in Merton, in the interests of time; the primary users of the Building Birth Certificate will be in Merton. The group's review of the collected data will provide clues to navigability problems in addition to any errors or bugs this system contains. For example, if users repeatedly click an image, the project group will alter that image to make it look less like a button in the next iteration. For a sample Usability Study Data-Sheet, refer to Appendix B.

Online Entry Form Pilot Test

Once the online entry form is completed, as shown by the usability study, it will be accessible from the Internet. The project team will then advise Adrian Hewitt to allow the entry form to be available to those who will need it in a pilot test. The test will allow the project team to find any final problems with the system as a whole. The types of problems that could arise are numerous and varied, ranging from hardware issues to database errors. As such, the Team will be prepared to deal with all issues that arise. If there are any problems the Team cannot solve without outside assistance, the Team will use every available tool to find a solution. The aforementioned tools include the Internet, customer support, and Information Technology professionals. The project team will continue to make small alterations to the form at the liaison's request until the liaison approves of it as a final product. This iterative process would occur over a period of several weeks.

Energy Reporting on the National Scale

The second goal is to set up the foundation for a system of reporting energy to the United Kingdom government. The Team will look into the current City Knowledge system in place in the Borough of Merton and see how this could integrate into the report. The project group, to find what information is missing, will also conduct interviews of the individuals receiving the reports. The Team will then incorporate these needs into a system to simplify the reporting process.

City Knowledge Assessment in Merton

In order to evaluate how to incorporate Building Birth Certificates into a City Knowledge system, the Team will research the City Knowledge implementation already in place in Merton.

- Gain access to current City Knowledge implementation.
  - Contact Merton liaison.
- Study current City Knowledge system in place in Merton.
  - Become familiar with what information is included in the Merton City Knowledge database.
  - Determine what information already in the Merton City Knowledge system applies to Building Birth Certificates and the Merton Rule.
- Construct appropriate data structures for Building Birth Certificates in the current City Knowledge implementation.

Energy Reporting Needs Study

The project group will conduct a needs study to determine what information is needed in the energy reports and appropriately represent the information.

- Determine to whom in the United Kingdom government the energy information will be reported.
  - Contact Merton liaison.
- Contact and interview the individuals receiving reports.
  - What information do they want specifically?
  - How do they want it formatted?
  - Should it be online vs. offline?
  - Are there other requirements?
  - Justify demands.

Energy Report Method Design

Based on the specifications above, the Team will design a system to allow for effective and efficient reporting of the energy data to United Kingdom government. The design specifications for the system will meet as many demands as possible.

Expansion Beyond Merton

The Merton Council would like to explore the possibility of expanding of the monitoring system outside Merton. The Third Team, working during the summer of 2008, will be responsible for constructing methods for other boroughs to adopt the system being developed in Merton. As the Team that developed the foundation of the system, the First Team will provide a set of guidelines to assist the following groups in this task.

Miniaturized City Knowledge

Concurrently with expanding beyond Merton, the Team will evaluate the possibility of miniaturizing the City Knowledge system creating a sort of "Building Knowledge." This smaller system would monitor sensors within a building and report the collected data to the Borough of Merton. Before beginning work on a smaller system, the Team will interview Adrian Hewitt and municipal building owners. If there is no interest in a "Building Knowledge" system, the Team will abandon this idea to focus on other tasks.

Data Collection Unit Specification

The Team will also draft a specification for data collection units. One data collection unit would reside in each Municipal building. The unit would be responsible for gathering data from various sensors and storing it, until Merton requests that the collection units upload the data to the Merton system. The draft specification will incorporate the proposed miniaturized City Knowledge system. The implementation of this proposal would ease data collection for municipal building operators since the data collection units automate data acquisition and reporting. The draft specification would include details such as system design, sensor interfaces, and reporting protocols.

Results

While in London, Team One established the foundations of the long-term project, which future teams will continue. To demonstrate the overall project, Team One set up a test bed inside the Merton Civic Center. Power meters installed onto the wind turbines at the top of the center fed into a computer acting as a server. This "server" then processed the data and made it available to any other computer on the network.

Team One constructed the program responsible for processing this information. This program also made it possible to share the information over the network. After further writing, the program became fully scalable to use at any level of data collection in the long-term project.

In the completed long-term project, planning agents will analyze the collected renewable and non-renewable data against predicted data. The planning process obtains this information from information the developer enters. This will include completing RELIEF-Web, a web-based form that requires contractors to fill out the building's specifications along with renewable energy initiatives. This form, also created by Team One, referenced the Building Birth Certificates previously used by the Planning and Projects Department.

In order to ensure continuity over the long-term project Team One also put together a set of guidelines for both the second and third teams. These guidelines are intended to give Teams Two and Three a better understanding of what they still need to accomplish to reach the final goal.

Interviews

The Team created a preamble and a list of sample interview questions to be used for the usability and needs study. Once in London, the Team created a new preamble to reflect the updated project (Appendix D). The original list of sample questions was also discarded due to their lack of relevance to the project. Below are a few contacts Team One made while in Merton and found useful to the project.

RELIEF

Gary Shaw

Gary Shaw is the GIS manager for Merton. He stated that the builder first applies for construction permission through the Development Control and Planning and Projects

departments; once the construction is finished, Development Control notifies Declan Stegner of its location. He then assigns it a Unique Property Reference Number (UPRN). These numbers are sequential and are assigned in the order completed (i.e. 48000101, 48000102, 48000103, etc.). The only set numbers in the Merton developments' UPRN are the "48" at the beginning to denote which council the building is located in. Each council is assigned a different classification number to improve organization inside the national GIS to which each council's GIS must report.

Kelly Howell

Kelly Howell is the Planning Systems Manager and the main representative for the 1APP system for Merton. Team One met with her to discuss the possibility of integrating RELIEF into 1APP. The basic layout of 1APP is that the national government requires only six points of information while the local offices can select from a list of information topics for what they specifically require. She felt that because of this, getting any information on the 1APP system pertaining to RELIEF would be nearly impossible.

Ed Cotterill

Ed Cotterill is a Masters Scholar at the Center for Alternative Energy and an asset to Adrian Hewitt in the creation of the original Building Birth Certificates. He had a list of additional fields that he found useful when looking into hard copies of the planning applications of buildings already built with renewable sources installed and working.

Reporting System and Civic Center Test Bed

David Bell

Dave Bell is the Energy and Sustainability Manager for the Civic Center. The Team wanted to set up a test bed for this using the turbines at the top of the Merton Civic Centre. Mr. Bell explained that a sensor could be applied to each wired pair, enabling the system to monitor two separate pairs of turbines. To acquire a computer he suggested contacting the IT Department to find an available computer that was out of use. He also provided a few contacts from which reliable metering technology could be found. He offered to fund this test from his department's budget when a suitable energy-monitoring device was found.

Declan Stegner

In order to continue research into the Civic Center test bed Team One contacted Declan Stegner to obtain a computer to use during the test of the turbines at the top of the Merton Civic Centre. The only necessary specifications were Windows 2000 or better and the ability to

connect to the Civic Centre network on the 15th floor. A computer was obtained from the IT storage department.

Russell Lerman

Russell Lerman is the main contact for Envido, a metering company already working with Merton. Team One met with him to discuss power meter possibilities for the Civic Center test bed using the wind turbines on the roof. Upon inspection of the turbines, he concluded that these were the wrong type to connect directly with a computer (the digital readouts on the units didn't pulse). He suggested the use of a few wireless monitors that could connect with a data logger he was installing on the sixth floor that would connect directly into the Internet. It was suggested to hardwire the sensors directly into a computer acting as a server; however, he thought this would not be as effective a method. Mr. Lerman also demonstrated the website end of his data logger system, which is very similar to the long-term project. It provides secured levels of access and organized by level (i.e. sensor, building, borough, city).

RELIEF (Renewable Energy Localized Installation Entry Form)

Team One's main task was to create a usable entry form that would allow any developer or planner to enter the information and submit it to the Planning and Projects Department in Merton. What the Team discovered upon arriving in Merton was that the planning applications did not use Birth Certificates; they were solely for Merton Rule compliance. Because of this, the Team changed the methodology to the following:

- Research the current planning permits process.
- Expand upon current Building Birth Certificate structure to create RELIEF
- Perform Needs Study to with the Excel version of RELIEF to gather information on possible problems, missing data, and potential loopholes.
- Design a web-based RELIEF. The Needs Study provides the content for the form.
- Perform a Usability Study on the web-based RELIEF to determine whether users are capable of using it.

1APP

Upon learning of the future 1APP system, the Team's initial idea was to find a way to have RELIEF added as a form provided by this online development application system. This would not only make the process simpler for the developer when filling out the applications, but

it would also make RELIEF a mandatory form to be completed before the building is approved for construction. Upon further research and consultation, however, the Team discovered that this would not be possible. Kelly Howell, the Planning Systems Manager, explained during an interview with the Team that only specific options could be placed on the 1APP system. Since it will be a National system, the specifications are assigned at a national level. Specific boroughs are allowed to require more information from the developers, but that information must be selected from a list provided by the national government. The Merton Rule is a borough initiative and not included on this limited list. RELIEF, therefore, cannot be included in the 1APP system. See Appendix F for the 1APP leaflet.

RELIEF and Merton Rule Enforcement

Since the 1APP system will not be able to enforce compliance with the Merton Rule, the Development Control Department will not be able legally to stop the developer's application process. However, upon approval of the application they can impose a few development conditions. One of these conditions states,

"No development shall take place until the Local Planning Authority has approved a report provided by the applicant identifying how the predicted CO_2 emissions of the development will be reduced by at least 10% through the use of on site renewable energy equipment. The developer shall install broadband/wireless remote sensors so that Merton Council (or an approved organization) can monitor the energy usage of the building, and/ or the performance of the renewable energy equipment. Before any unit is occupied or sold the renewable energy requirement shall have been installed and the local planning authority shall be satisfied that their day to day operation will provide energy for the development for so long as the development remains in existence." (Appendix U).

This means that although the Merton Council cannot stop the application process, they can prevent the developer from beginning construction until it is satisfied with their on site renewable energy sources. The developer is also obligated to install a monitoring system to the council's satisfaction, enabling the borough to ensure that the new development is complying with the Merton Rule before it can be used.

Start to Finish Process for RELIEF

Combination of Information

The current Building Birth Certificate that Adrian Hewitt uses works fairly well as tool for the Planning Department to use. He is interested in an online system that would allow him to review the information filled out by developers. It is for this reason that an extensive redesign of the form had to occur. The Team met with Ed Cotterill to gather any other information he had found in his recent work. Incorporating his suggestions and Adrian Hewitt's Birth Certificate, the Team created the first draft of RELIEF. RELIEF is the system encompassing the data requested in the Building Birth Certificates, a web-based data entry system, and the database, which stores the collected data. This draft used Microsoft Excel to create a visual representation while the database was in development, as shown in Appendix H.

RELIEF Needs Study

The Team sent a copy of the original Building Birth Certificate to a list of contacts provided by Adrian Hewitt (Appendix G). This list was constructed to include those that this form would affect the most. Developers, companies, colleagues of Adrian Hewitt's, and his counterparts in other boroughs were all included. A few of the replies are below.

- Would individual boroughs be responsible for collating the information?
- Build a database of existing sustainability features on developments that have gone through the planning and building control process - i.e. mapping what was already out there.
- It would be great if anything you come up with /develop could be populated retrospectively. – The Team has addressed this.
- It would be useful to include other sustainability measures e.g. gray water recycling, green roofs, etc. [*sic*]
- Would we be able to tie this monitoring system in with the Code and BREEAM certificates and post construction reviews to help keep track of when developments should submit these reports?
- The table makes reference to the Code but not BREEAM for non residential / mixed developments
- Would it also be helpful to refer to district wide heating/CHP/Private wire within developments as they could link to other developments later?

- It might be useful to reference the fuel types used in developments, which are using biomass/Bio fuel? (This seems to be a very popular technology for offsetting large quantities of carbon. This reference could help monitor the fuel demand in London and perhaps the setting up of further suppliers/look at the sustainability of this option i.e. where their fuel source is.)

Creation of RELIEF

With the draft of RELIEF and the comments received from the consultations, the Team developed a preliminary version of the web-based RELIEF, referred to as RELIEF-web. RELIEF-web contains three core elements. First, the developer enters information about the development as a whole. This information includes the address, postcode, and developer name. Second, the developer adds individual components to the development. For example, if the developer envisions a building, which has a bank, a fast-food restaurant, and a gym, he would enter the details for each during this step. As the developer adds each component, RELIEF-web calculates the expected energy use and carbon emissions of the overall development. Finally, the developer provides the details of the renewable sources used in the development and any additional energy efficiency measures. With the predicted carbon emissions from the previous step, RELIEF-web also indicates achievement of the carbon reduction guideline. The interface for this system is minimal; this task remains until after the usability study.

Once the developer completes RELIEF, the system compiles and stores the data into a database. Although not yet implemented, RELIEF will link with Merton LOUIS, allowing the GIS to display the carbon baselines for each development. Combined with the reporting system, RELIEF will provide a powerful tool for tracking each development's carbon emissions and Merton Rule compliance.

Source Data for CO_2 Calculations

The data for RELIEF's built-in energy calculator comes from the London Renewables Toolkit[32]. This document combines energy benchmarks from four sources: the Energy Consumption Guide, the Energy Efficiency Office, the Introduction to Energy Efficiency, and the

---

[32] (Maunsell, 2004, pp. 107-109)

BMI Special Report: Energy Benchmarking in the Retail Sector. RELIEF also incorporates the baseline carbon emissions provided by Adrian Hewitt.[33]

RELIEF-web Usability Study

To determine the usability of the new RELIEF prototype, the Team sent it to the same group of contacts used for the needs study (Appendix J) This was to ensure continuity of contacts, ensuring the responses remain relevant to the previous suggestions. The process of receiving responses was simplified for this test. Instead of waiting for email replies, as was done during the needs study, a comments section was added at the end of the RELIEF. Using an administrator's view of the form, the comments will be viewable only by the Team. Some comments are below.

- Michael Carnuccio, michael.carnuccio@southwark.gov.uk

  - "Potentially a very useful tool for estimating energy uses and carbon reduction from proposals. However, I could not test it properly due to an XML error every time I completed the form and tried to add it to my development. I think it would be helpful to explain exactly what is meant by some of the fields/what type of data is expected to go in. E.g., what does the field Building Regulation mean? Also as roof type, what about adding green roof in, as these have energy saving impact. In addition, should we now be using 20% target as per London Plan? [*Sic*]"

Finalized Live Version

The Team has finalized RELIEF so that it may be launched on the LOUIS server. This final live version will allow developers to use the form outside of a test model and the Plans and Projects Department to benefit from the information. Dr. Carrera has been contacted to place RELIEF on the server.

National Energy Reporting

The secondary goal of the C term 2008 project team was to begin work on a prototype software system for reporting energy on a national level.

---

[33] The source of these figures, Building Regulations Approved Document L2, is not available for reference

Design Considerations

Users of the System

The project team concluded that the system has two distinct groups of users: those looking at the energy reports, and those who have to modify it in order to keep the software updated with the latest hardware. The Team decided that the final viewing system would be web-based or GIS based, and thus data would need to be stored in the form of a database. In addition, the system would need to allow for any type of data sensor, regardless of make, model, or method of input. The Team concluded that the system needed a database-interfacing module, and a plug-in module.

Purpose of the System

The Team concluded that the purpose of the overall system is to collect data and send it to the government in an acceptable format. In addition, the system would need to be able to send the data across a network. The Team concluded that the system would need to include a networking module, and that the plug-in module was essential.

City Knowledge Principles

The Team then began brainstorming about the system from a City Knowledge perspective. The Team analyzed each of the City Knowledge principles to determine how to incorporate them into the system. The Team concluded the following:

- The system itself was a representation of the Middle-Out Approach
- The system's network-like structure with individual nodes adheres to the Atomic Data Structure principle.
- The system's database and sensor feedback systems adhere to the Sustainable Update Mechanism principle
- The system's ability to alter a web page based on sensor input, in addition to reporting data to an authority, adheres to the Sharing of Information principle.
- The system's plug-in architecture allows it to gather any kind of data, not just energy usage, which different departments can use. This mechanism is a representation of the Interagency Coordination principle.
- The Team concluded that each building or complex would have a system of its own gathering the data locally, and thus, would own it. This consideration is representative of the Distinct Informational Jurisdictions principle.

The Team relayed this information to Dr. Fabio Carrera. Dr. Carrera approved of the idea, agreeing that for the reasons listed above, the system does adhere to the City Knowledge principles.

Simplicity

The Team then brainstormed about ways to simplify the system, and thus make it more stable. The Team concluded that the system should require absolutely no keyboard input. The only input required by a user should be mouse input. All other information needed by the system will be in the form of a configuration file, or the system will request it from a system further up the chain. As such, the Team concluded that at a lower level, the system will only have a mouse and would disable most operating system functions. In addition, the system would require an automatic update mechanism, which would eliminate the need for a dedicated team of support and update specialists.

Security

The Team then discussed the concept of security. The Team concluded since both a governing body and private companies would use the system, it required some form of security. The Team then discussed the concept with the Merton liaison, Adrian Hewitt, and he agreed that security is vitally important. The Team developed the security recommendations seen in Appendix N.

Overall System

The Team came together and discussed what the separate components of the system would be. The Team concluded that the prototype computer would have the following software:

- Windows Server or Windows XP
- Microsoft .NET Framework 2.0
- Apache 2.2 (A web server)
- MySQL 5 (A database server)
- PHP 5.2.3 or newer

The developed software system would require the Microsoft .NET Framework 2.0, which will run on Windows XP. Apache 2.2 and PHP 5.2.3 are for website-purposes, and MySQL 5 is both for the website and for the program.

The Team then discussed the software system in detail. The Team concluded the software system's components would reflect the considerations taken above, and the final program would have:

- A Networking module
- A Plug-in module
- A Security module
- A Database module
- An Updater module
- A Sensor module
- A Data Analysis module
- A Simple User Interface

Merton Rule Hierarchical Online Monitoring and Energy Reporting (MrHomer)

The Team started the software system with MrHomer, a framework that would perform the more in-depth and software-intensive tasks of the system. The Team designed MrHomer with reusability and flexibility in mind, as well as simplifying the final program. The MrHomer system contains the crux of the networking module, the plug-in module, the security module, and the database module. In addition to extensive testing, the Team fully documented the system, including a user guide, tutorials (Appendices O, P), a software architecture document (Appendix M), and diagrams (Appendix Q).

Energy Reporting Backend

The second layer, or application layer, of the energy reporting system hides the functionality of the framework and provides the features needed for energy reporting. This layer incorporates the security protocol selected for the system (covering encryption and authentication), holds the database structure used, and handles detection of sensor interfaces. Designing the system in this way isolates the program from changes to the underlying system, such that switching database engines will not affect operation the system.

Database

The application layer provides engines to interact with different database storage mechanisms. From the perspective of the program, these engines are interchangeable. The Team has provided two such engines: one for Microsoft SQL Server, and one for MySQL. The Team used Microsoft SQL Server for development and testing; the final implementation will use

MySQL. This change is transparent to the program, provided both databases have the same structure.

Communication

The communication component integrates the security and networking components of the framework. This component handles accepting incoming connections, making connections to other systems, authenticating systems, and encrypting network traffic.

Sensors

Using the plug-in system provided by the framework, the application layer provides a mechanism to manage sensors. This component automatically detects new types of sensors and loads them into the program. When the program detects a new sensor installed in the building, it detects the type and loads the proper interface code. This feature allows the system to add or remove sensors while the program is running.

Application Updater

The application updater receives messages from systems higher in the tree when new updates are available. This includes new sensor types, a new security protocol, or any fixes to the program itself.

Civic Center Test Bed

To continue informing his colleagues, superiors, and those interested in the Merton Rule City Knowledge Project, Adrian Hewitt has been giving many presentations throughout the UK. In order to assist him and make the project more comprehensible, the Team decided to create a simple test bed. This involved monitoring energy produced by the wind turbines fixed atop the Merton Civic Center as well as a portion of the non-renewable energy consumed by it. Sensors capture these energies and report them to a "server," a nearby computer, and processed by a program there. After a quick computation, the system makes data available to anyone in the network. Upon access the user would see the comparison between the two energy sources and whether the Merton Rule was being met.

This test bed remained as simple as possible to display the general goal of the project. The second Merton WPI team will find appropriate sensors for the necessary information and a suitable network for access outside the Civic Center. It will be the job of the third team to compile the previous projects into a substantial form for Adrian Hewitt to present to the government officials responsible for funding the actual project based on this prototype.

Unfortunately, Team One was unable to complete the test bed due to lack of time and equipment. Although the Team was able to secure a server computer, a monitoring device was much more difficult to obtain. After a meeting with Russell Lerman, a representative from the monitoring company Envido, prices for equipment were forwarded through email to the members of Team One. At £1550.00, the price was too steep to continue and not enough time remained to begin research into another company. Since this test bed will be a valuable asset to Adrian Hewitt when explaining the entire project, Team Two will continue work to complete it.

Research Performed by Team One for Continuing the Project

Methods of Networking

Team One considered a few possibilities for the networking of the city knowledge system to reach every building in the target area. Future teams must further research the best method of transferring the data.

- Broadband would provide a quick connection between all users, but Team Two would need to find a company to allow access. Although quicker, broadband is not available everywhere and would leave many buildings out of the network, reducing scalability.
- CCTV (Closed Circuit Television) is in use throughout Britain.[34] This does provide hardwire access to the majority of UK buildings, but it would require the permission of every owner and operator of the cameras. It would also reduce scalability outside of Great Britain because of limited CCTV use.
- BT Phone Lines – Nearly every building in all developed countries contains a landline. This would provide slower, but universal access to nearly every building in not just Great Britain, but most countries outside it as well. The Team would need to contact a phone company to request the use of their lines. BT (British Telecommunications) seeks to lead other phone companies in the field of carbon reduction.[35] This is a great place to start for this type of research.

Hardware Research

A previous Interactive Qualifying Project done regarding electricity usage on the Worcester Polytechnic Institute campus researched into hardware capable of monitoring this

[34] (Privacy International, 1197)
[35] (British Telecommunications, 2008)

information.[36] In addition to consulting this IQP, Team One conducted a small amount of research with UK based companies (Appendix E).

London Borough of Southwark

The Team went to The London Borough of Southwark on Friday, February 15, 2008 to discuss their involvement with a research project on a wind turbine located on top of a building in the borough. The focus of the project is to obtain real data of energy production in an urban environment. The Team was interested in how the group was monitoring the energy and the types of equipment they were using. The Team discovered that the system was quite simple and easy to operate. One problem was that the entire system was local, thus the researchers had to actually go out to the building and gather the data. Once the Team discussed the reporting project with them, they were very interested in the project and look forward to seeing what direction it takes.

Possible Test Bed Companies

Finding companies or residences that are willing to help test the City Knowledge energy reporting system during development would be beneficial. This would allow the discovery of real-time hitches hidden during the theoretical conception.

The Big Yellow Self Storage Company

The Team held a meeting with Nigel Hartley, the Construction Director of the new Big Yellow Self Storage Company building, on February 8, 2008. As the representative of the Big Yellow Group, he is interested in the idea of energy monitoring in Merton. He is willing to work with the Merton Government and WPI IQP Teams to test the City Knowledge energy reporting system.

---

[36] (O'Hara, Hobson-Dupont, Hurgin, & Thierry, 2007)

Recommendations

Guidelines for Team Two (D Term)

Team Two will have five main tasks during their time in Merton. Team One suggests that Team Two research them in this order.

- Test Bed Completion
  - Secure simple, inexpensive monitoring equipment
  - Compile with program and computer from Team One
- Networking
  - Research best method for relaying data collected by the sensors
  - Prototype program developed by Team One
    - Team One's program runs via the Internet
- Onsite Server
  - The computer/device that will relay the information from the sensor to the local borough
  - Requirements of the systems
    - Inexpensive
    - Be able to run Team One's program (See Software Architecture Document not finished yet)
    - Able to connect to Merton via Network
  - "Plug-ins" for sensors (Appendix P)
- Sensor Research
  - The second Team must determine the best sensor or installation on site. The following data inputs are required for the Merton monitoring system
    - All Renewable Energy/Heat used onsite
      - Electric outputs, heat outputs, amount of bio-fuel used, etc.
    - All Non-Renewable Energy/Heat/Water used onsite
  - The framework (developed by Team One) will support any data input wanted
- Finish Team Three's Task list to ensure project completion

Guidelines for Team Three (E Term)

- System Test Bed
    - Whole program running with all data inputs being monitored and reported
    - Integrate RELIEF and Reporting System
- Finalize User Interfaces for RELIEF and Monitoring System
    - The forms and displays must be user friendly – i.e. color-coding, relevant fields together, informative graphics, etc.
- GIS
    - Linking RELIEF to the GIS with the use of UPRNs
- Final Usability Study
    - Entire System Test Bed
- Presentation
    - Create a presentation that will demonstrate the entire system (Team 1-3 work)

Conclusions

RELIEF

With the completion of both the needs study and the usability study, the Team was able to complete RELIEF. The Team provided Merton with a working form for data collection, including the ability for developers to edit their renewable sources. The task of improving the appearance of the form remains for Team Three. The Team also provided an installation package for RELIEF, which will allow Merton to configure the system on a proper server with little effort.

Reporting

The Team also completed a prototype system for national energy reporting. The prototype covers all requested aspects of the system: energy monitoring sensors, reporting data to Merton, and reporting data to the Greater London Authority. Based on the prototype, the Team compiled recommendations for the final system. The Borough of Merton will provide these guidelines to a government-funded programming team to aid the development process. These recommendations are included in Appendix W.

Test Bed

Team One decided it would be an excellent idea to create a local test bed of the energy reporting system being developed. This would enable the current and future teams to observe any problems in the system that might have been previously overlooked.

The wind turbines mounted at the top of the Merton Civic Center were selected as ideal because of their ease of access and availability. David Bell gave the Team access to the monitors already in place and explained how the turbines were connected to the buildings wiring.

After meeting with the representative of Envido, a single device with two pulsing outputs was selected. This would allow the Team to use two different monitoring strategies and select which worked best. When the email arrived from Envido, detailing the price of installation, the Team decided that the test bed could not be completed during that project. £1,550.00 was too much money to spend on a test system and not enough time remained to continue research into another company.

Since this test bed would be an invaluable asset to Adrian Hewitt when explaining the proposed energy reporting system, Team Two will complete it during the following Merton IQP.

Appendix A: Merton Background

The Borough of Merton is one of the 32 boroughs of Greater London. It is located south and west of the city of London. Merton formed from the merger of Mitcham, Morden, Wood, and Wimbledon in 1965. Since then, Merton has flourished into a self-sufficient community that includes many commercial shopping centers. Merton is comprised of 20 wards.[37]

Figure H: Location of Merton with the Boroughs of London

Demographics

Merton is home to approximately 190,000 citizens in a 9,380-acre area. Approximately 75.2% of the population is of white heritage. Asian and black ethnic groups follow up with 11% and 7.6% respectively. The average age of the Merton population is 36.6 years. About 63% of the borough identifies itself as Christian, followed by the 16.6% with no religious affiliation, with the remainder being of varied religious affiliations.[38]

Merton Council

Demographics

Every year, each of the twenty wards within Merton elects three representatives to the Council of Merton. These councilors represent Merton's three political parties: Conservative (30 councilors), Labor (27 councilors), and Merton Park Ward Independent (three councilors). The council then selects nine of the newly elected councilors as cabinet members, and three as assistant members. The cabinet appoints one of its members as Leader of the Council. Finally,

---

[37] (The Council of Merton, 2007)

[38] Ibid.

the council selects three more members as Chief Executive, Assistant Chief Executive, and Mayor.[39]

Budget

Major sources of Income

The 2007-2008 Guide to the Council Tax reveals the planned expenditure of the Merton Council to be £423,001,000. Over two thirds of the planned income is directly from the city services themselves, with the remainder making up the actual taxes (Figure K). Furthermore, the taxes themselves subdivide into three categories: Non-Domestic Rates making up 38%, Revenue Support Grant making up 6%, and Council Tax on residents, making up 56%.[40]

Tax-based Income

Legend:
- Non-Domestic Rates
- Revenue Support Grant
- Council Tax Income
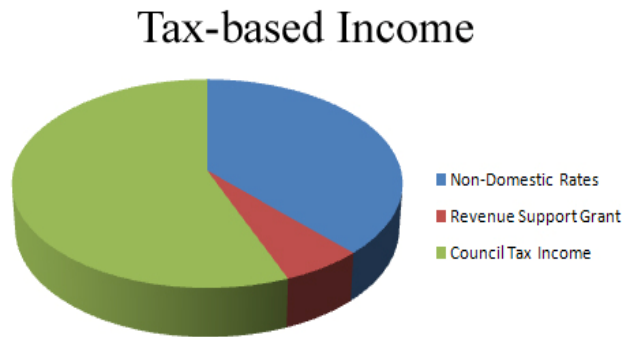
Figure I: Merton Council – Tax-Based Income

---

[39] (The Council of Merton, 2007)

[40] Ibid.

Department Income
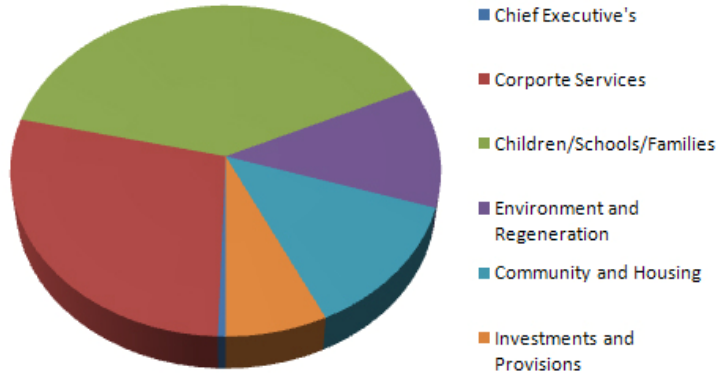
Figure J: Merton Council – Department Income

Total Income
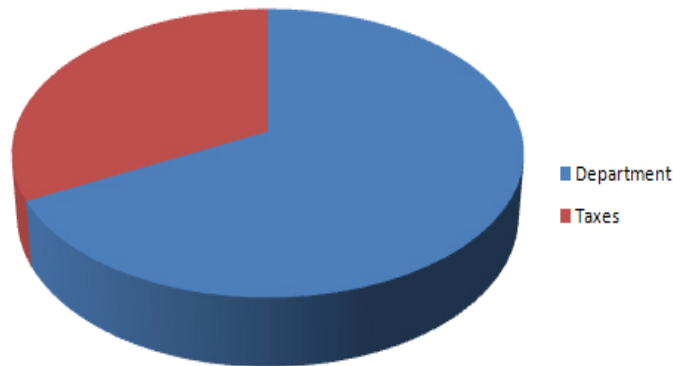
Figure K: Merton Council – Total Income

Budget Plan

The Council plans to spend £423,001,000 in the 2007-2008 fiscal year. Over one third of the budget is going directly into schools and other services to benefit children and families, followed by Corporate Services (albeit by a £60,000,000 margin). Closely following are Community and Housing, and Environment and Regeneration (Figure L).[41]

---

[41] (The Council of Merton, 2007)

Planned Gross Expenditure

- Chief Executive's
- Corporte Services
- Children/Schools/Families
- Environment and Regeneration
- Community and Housing
- Investments and Provisions
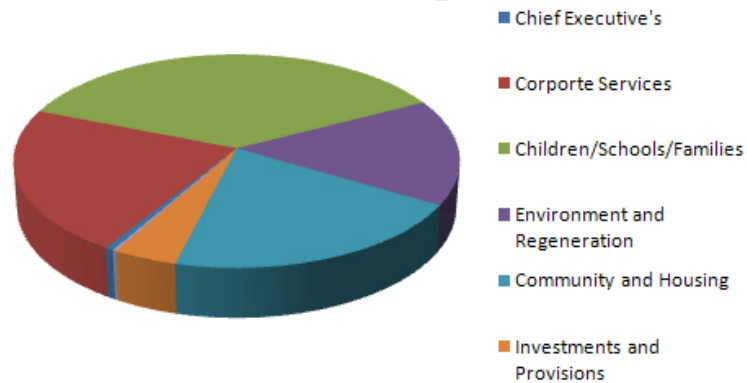
Figure L: Merton Council – Planned Gross Expenditure

Organization

The Council of Merton houses five departments, each responsible for distinct tasks. This allows the Council to divide its workforce and to allocate resources as required (Figure M). The first of these, the Chief Executive, advises the remainder of the council on strategies and policies for services. Corporate Services, the second department, interfaces with visitors, provides technological support, and catalyzes a variety of other services. The Children, Schools, and Families Department ensure quality education and healthcare for children and families. The Community and Housing Department seeks to develop and repair housing, especially for those in need, and to provide social care services.[42]

---

[42] (The Council of Merton, 2007)
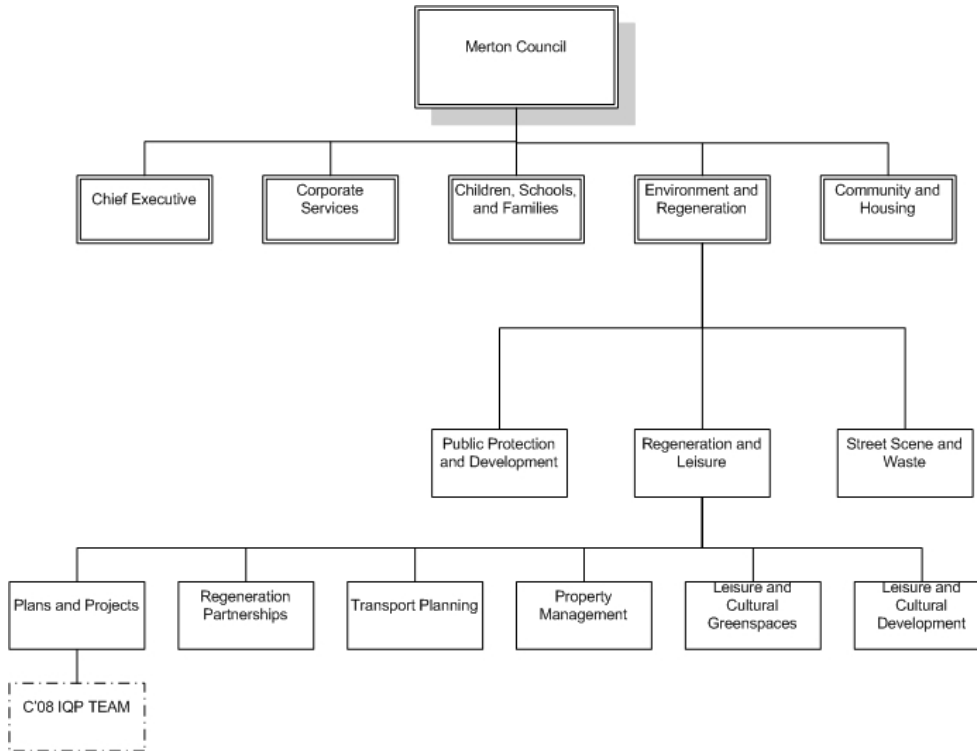
Figure M: The Council of Merton Organization

Environment and Regeneration

The final department in the Council of Merton is Environment and Regeneration. The department maintains the local environment and economy and seeks to improve these. The Environment and Regeneration Department is responsible for many of the zoning and permit laws, as well as the Merton Rule. As shown in Figure M, it comprises three divisions: Street Scene and Waste, Regeneration and Leisure, and Public Planning and Development.

Street Scene and Waste

The Street Scene and Waste division is similar to the highway departments of many local governments. It is responsible for maintaining the quality of roadways, keeping them clean in the winter, and servicing them as required. It also designs and implements traffic calming techniques and other strategies to further the safety of motorists, cyclists, and pedestrians alike. It is also responsible for providing specialized transport for the disabled. However, with its simpler duties, it is also tasked with providing garbage and recycling services, as well as maintaining all the landfills in the borough.

Regeneration and Leisure

The Regeneration and Leisure Department consists of six teams: Regeneration Partnerships, Transport Planning, Property Management, Leisure and Cultural Green Spaces, Leisure and Cultural Development, and Plans and Projects. Regeneration Partnerships works mostly with helping those in poverty-stricken areas of the borough. Transport Planning updates transportation policies attempting to keep the roads safe for both motorists and pedestrians. The Property Management team maintains the records and controls the assets of the Merton Council. Leisure and Culture Development encourages the citizens of the borough to partake in healthy physical exercise while Leisure and Cultural Green Spaces works to promote the exploration of the environment. The last team, Plans and Projects, contains the current projects underway under the supervision of the Merton Council. Plans and Projects is also responsible for the application and use of the Merton Rule, the rule stating that all buildings need to cut Carbon Dioxide emissions by 10% by using on-site renewable energy sources.

Appendix B: Sample Usability Data Sheet

Name:		Anonymous:	
Age:		Computer Experience:	
Profession:		Visual Impairments:	

Task 1	
Time Taken:	
Issues:	
Other Comments:	

Task 2	
Time Taken:	
Issues:	
Other Comments:	

Task 3	
Time Taken:	
Issues:	

Other Comments:	

Post-Trial Interview:

Likes:

Dislikes:

Suggestions for Improvement:

Comments:

Appendix C: Sample Interview Questions

Preamble

We are students at Worcester Polytechnic Institute in Worcester, Massachusetts. We are conducting a research project on behalf of The Borough of Merton to gather information pertinent to a Building Birth Certificate online data entry form, which will be used for compliance with the Merton Rule in the Borough of Merton. As part of this project, we are conducting a series of interviews with key individuals. We have asked you to participate because we believe you have unique knowledge of these issues that will be valuable to the project.

Before we begin, we would like to thank you for taking the time to participate in the interview. Your participation is voluntary. You may refuse to discuss any question or terminate the interview at any time. With your permission, we would like to record the interview. The tapes, notes, and subsequent transcripts of the interview will be kept confidential, and will be accessible by only the members of the Team and our immediate faculty advisors. Your name will not be used in any subsequent report or publication without your permission.

If you consent to be interviewed at this time, we would ask that you indicate your agreement below.

I agree to participate in the interview _____   _____

Interviewee Signature   Date

_____

Interviewee Name

Please initial for permission to record _____

Interviewee Initials

_____   _____

Interviewer signature   Date

Online Entry Form Needs Study Sample Questions

- What information are you looking for that is not currently a part of the Building Birth Certificates?

- Why do you need this information?

- How does this coincide with the Merton Rule?

- Which information is the most important? (List according to importance.)

- What measurement units do you want set as the standard? (i.e.: kips/ft, kilograms/meter, megawatts?)

- Will there be a required standard of this information to allow the construction of the building? If so, what is this standard?

- Who will be responsible for entering this information?

- Will there be a way to find this information from older buildings as well?

- Current Building Birth Certificates: Online or Offline?

  o Is a web page needed for entry of all information from Building Birth Certificates? (Not just Merton Rule info?)

  o Would the data need to be sectioned by relevance? If so, how would it be split up?

- Will this information be placed in the City Knowledge system? (Unless it is already)

  o Direct incorporation into CK through website?

Energy Reporting Needs Study Sample Questions

- What information do you need specifically?

- Why is this information necessary?

- Will all of this information be available through the improved building birth certificates, or do you foresee a possible need for a secondary system?

- How do you want this data formatted?

- Do you want an online reporting system or would you prefer hard copies of all data reported?

- How will this information improve upon the Merton Rule idea? (Better enforcement, more efficient?)

- How often will this information need to be reported? Monthly? Annually?

- Since this information will be incorporated with the information from the rest of the nation to report to the United Nations, what are the demands of the U.N.?

    o How do they want it formatted?

- How do they need it reported? Online/Offline?

Appendix D: Preamble

IGSD Merton IQP Team C Term 2008   The Council of Merton

Worcester Polytechnic Institute   Planning and Projects Department

100 Institute Road   Merton Civic Centre, London Road,

Worcester, MA, USA 01609-2280   Morden SM4 5DX

We are students from Worcester Polytechnic Institute in Worcester Massachusetts, USA. We are currently conducting a research project in partnership with the Planning and Projects Department of the Merton Council to develop an energy information reporting system to be used throughout the borough. This information will be used to monitor the amount of renewable energy being employed by each building, generated by onsite sources, to determine compliance with the Merton Rule. The system will be based upon the City Knowledge concept, which is a set of principles, developed by Dr. Fabio Carrera that governs the storage and accessibility of information. To ensure the correct data is collected, a preliminary "building birth certificate" must be completed online by the building contractor dictating the building's specifications as well as how it will obtain its energy (renewable, non-renewable). It is our hope to eventually incorporate this system into other boroughs in London, and throughout the UK.

We would like to sincerely thank you for your time and contribution to this project.

Robert Bernard

Benjamin Lessard

Brian O'Keefe

Christopher Scalabrini

Appendix E: Metering Solutions

Appendix F: 1APP Leaflet

Communities and Local Government is launching 1APP, a Standard Planning Application Form to replace all existing types of planning application forms (except minerals) within England.

Currently, Local Planning Authorities (LPAs) produce their own local planning application forms. These often differ creating inconsistencies in the planning process.

1APP will remove the differences in application forms by ensuring the same information will be required for comparable applications by every LPA in England. This standardisation will make the planning system clearer and more efficient for planning professionals and citizens alike.

1APP is primarily designed as an electronic form and will be available online via the Planning Portal **www.planningportal.gov.uk** – however it is also available as a paper form from LPAs.

How 1APP will affect you

The various planning application forms currently in use by LPAs differ in the questions asked, their style and the number of copies of the form needed for submission.

This inconsistency can create difficulties for applicants, particularly those making multiple applications, sometimes to different LPAs.

1APP will save applicants time and effort completing applications. The new form will make the application process easier by asking standard questions and ending confusion over individual LPA requirements and offering guidance and help at every stage.

1APP in its online form will intelligently determine the planning consent type required, and specify the LPAs additional information requirements, significantly increasing the chances of a valid application being submitted first time. This benefits both the applicant and the LPA staff who process the application, potentially resulting in a faster decision.

If you're a planning professional:
You will benefit from

- a simpler, streamlined application process
- familiarising yourself with just one form for all your applications
- saving standard information (e.g your address and contact information) so you needn't re-enter it for every application

If you're a member of the general public:
You will benefit from

- a simpler more transparent planning process
- clear online guidance where and when you need it during the application process
- a better chance of submitting a valid planning application

Timescales

1APP will be available from February 2007 and will become the only legal way of submitting a planning application within England on 1st April 2008 (except for minerals).

Some LPAs may even prefer to accept the standard application form as early as February 2007, so check with your LPA before completing your application.

www.planningportal.gov.uk/1app

Appendix G: Original Building Birth Certificate

MERTON RULE BIRTH CERTIFICATE	ANSWERS				
	DROP DOWN MENUES				
GENERIC FIELDS CAPTURED BY DC					
UPRN	No				
Address	Yes				
Developer	Yes				
Originator DC Officer	Yes				
Development Control registration number	Yes				
Building type : **Yes**					
Occupation ownership : **Yes**					
Size m2: **Yes**					
Dimensions m2 - **Yes**	Volume	Floors	Hight	Width	Length
Part of larger development? **Yes**					
If yes give details: There would have to be an underpinning Acess flat programme for mixed developments					
	Type of development			Size m2	Res units
Development mix details					

FIELDS NOT CAPTURED		
Code for Sustainable Homes **No**		
Construction method : **No, but probably BC**		
Heating sytem installed No, but probably BC		
Number of windows (external) **No, but drawings would show this detail.**		
U values **No**		
Total window area (m2) : **No**		
Roof insulation standard : **No, but probably BC**		
Wall insulation standard **No, but probably BC**		
Air conditioning : **No, but probably BC**		
Occupation activity		
Geology (if other give details)		
Surrounding land (Garden/Car Park/Yard)	Type	Area

UNIQUE FIELDS					
Predicted heat energy (kWhpa)		$CO_{2\,kg}$	0		
Predicted electric energy (kWhpa)		$CO_{2\,kg}$	0		
Renewable Energy (type 1)					
Product installed (1)		Unit	Amount	kW	$CO_{2\,kg}$
Renewable Energy (type 2)					
Product installed (2)		Unit	Amount	kW	$CO_{2\,kg}$
Renewable Energy (type 3)					
Product installed (3)		Unit	Amount	kW	$CO_{2\,kg}$
Renewable Energy (type 4)					
Product installed (4)		Unit	Amount	kW	$CO_{2\,kg}$
Total renewable energy/CO2					0

Additional thermal efficiency measures	Measure	Amount	kW cut	$CO_{2\,kg}$
1				
2				
3				
4				
5				
6				
Total renewable energy/CO2 reduction				

Additional electrical efficiency measures	Measure	Amount	kW cut	$CO_{2\,kg}$
1				
2				
3				
4				
5				
6				

Orientation of building (Front entrance)		
Total roof area		
Area of flat roof		
	Orientation	Area
Orientation of pitched roofs		

Appendix H: RELIEF Excel

RENEWABLE ENERGY LOCALIZED INSTALLATION ENTRY FORM		ANSWERS	*(+) = Multi Fields	Not Shown on Form
		DROP DOWN MENUES	Auto Show Fields	

Development General Information

UPRN	Number			
Date of Application	Text			
Address	Text			
City	Text			
Postcode	Text 3-3			
Council	Text			
Architects Name	Text			
Developer	Text			
Planning Consultant's Name	Text			
Mechanical Engineer's Name	Text			
Environmental Consultant's Name	Text			
Originator DC Officer	Text			
Development Control Registration Number	Text			
Development Status				
Development Type				
Development mix details		Type of development	Number of Units	
			Int	*(+)

Unit General Information

Occupation Ownership	
Occupation Activity	Text
Building Type	
Code for Sustainable Homes	
Ecohomes Rating	Int
Building Regulations Used	Text

Number of Windows (External)	Int
Total Window Area	(m2)
Floor Space (Bounded by Outside Walls)	(m2)

Dimensions	Floors	Volume	Height	Width	Length
	Int	(m3)	(m)	(m)	(m)

U Value	Roof	Ceiling	Walls	Floor

Air conditioning	(Make/Model)
Heating sytem installed	

Construction method	
Geology	

Surrounding Land	Type	Area
		(m2)	*(+)

Unit Energy,Gas and Water Predicted Usage Information

Unit Predicted Gas Usage	(kWhpa)
UnitPredicted Electric Usage	(kWhpa)
Unit Predicted Water Usage	Volume
Unit Total Energy Usage	Totals

Unit Gas CO2	(Kg)
Unit Electirc CO2	(Kg)
Unit Water CO2	(Kg)
Unit Total CO2	Totals

Development Renewable Energy Information							
Renewable Energy (Type 1)							
Product Installed (1)	Make/Model	Electrical Power (kW)	Electrical Annual Output (kWh)	Area (m2)		CO_2 (Kg)	*(+)
Renewable Energy (Type 2)							
Product Installed (2)	Make/Model	Electrical Power (kW)	Electrical Annual Output (kWh)			CO_2 (Kg)	*(+)
Renewable Energy (Type 3)							
Product Installed (3)	Make/Model	Electrical Power (kW)	Electrical Annual Output (kWh)	Heat Power (kW)	Heat Annual Output (kWh)	CO_2 (Kg)	*(+)

Total renewable Heat/Electrical Energy		Total	Total	Total	Total
Total Renewable CO2 Reduction					Total

Additional thermal efficiency measures	Measure	Amount	kW cut	CO_2	
1					*(+)
Total renewable Energy/CO2 Reduction				Totals	

Additional electrical efficiency measures	Measure	Amount	kW cut	CO_2	
1					*(+)
Total renewable Energy/CO2 Reduction				Totals	

Orientation of building (Front entrance)			
Total roof area	(m2)		
Roof Type			
	Orientation	Area	
Orientation of pitched roofs		(m2)	*(+)

Development Energy,Gas and Water Predicted Usage Information

Predicted Gas Usage	(kWhpa)
Predicted Electric Usage	(kWhpa)
Predicted Water Usage	Volume
Total Energy Usage	Totals

Gas CO_2	(Kg)
Electirc CO_2	(Kg)
Water CO_2	(Kg)
Total CO_2	Totals

Appendix I: Development Information Database

Section 1 – Planning Details

- Planning Application Number
- Address
- City
- Postcode
- Council
- Date of Application
- Development Status – In Planning / Approved / Under Construction / Completed
- Development Type – Residential / Commercial / Mixed
- Building Type – Conversion / New Build
- Part of Larger Development
- Developer's Name
- Architect's Name
- Planning Consultant's Name
- Mechanical Engineer's Name
- Environmental Consultant's Name
- Development Control Officer's Name

Section 2 – Building Details

- Size (m^2) / Number of Units
- Site Area (m^2)
- Number of Stories
- Height
- Building Regulations Used
- Eco-Homes Rating – (if used)
- Code for Sustainable Homes Rating – (if used)
- U Values
    - Walls
    - Floor
    - Ceiling

- o Windows

- Roof Type – Flat / Pitched

- Roof Area (m^2)

- South Facing Roof Area (m^2) – (if pitched)

Section 3 – Systems

- Air Conditioning

- Heating System

- CHP?

- Predicted Gas Usage   (kWh/annum)

- Predicted Electricity Usage   (kWh/annum)

- Predicted Gas CO_2 Emissions   (Kg/annum)

- Predicted Electricity CO_2 Emissions (Kg/annum)

- Predicted Water Usage   (m^3/annum)

- Predicted Water CO_2 Emissions   (Kg/annum)

- Total Energy Usage   (kWh/annum)

- Total CO_2 Emissions   (Kg/annum)

Section 4 – Renewable Solutions

- Total CO_2 Saving Required   (Kg)

- Photovoltaic

  - o Area   (m^2)

  - o Power   (kW)

  - o Annual Output   (kWh)

  - o Annual CO_2 Saving   (Kg)

  - o Cost

- Solar Thermal

  - o Area   (m^2)

  - o Power   (kW)

  - o Annual Output   (kWh)

  - o Annual CO_2 Saving   (Kg)

  - o Cost

- Wind Turbine

- o Power   (kW)
- o Annual Output   (kWh)
- o Annual CO_2 Saving   (Kg)
- o Cost
- Ground Source
  - o Power   (kW)
  - o Annual Output   (kWh)
  - o Annual CO_2 Saving   (Kg)
  - o Cost
- Biomass Heat
  - o Power   (kW)
  - o Annual Output   (kWh)
  - o Annual CO_2 Saving   (Kg)
  - o Cost
- Biomass CHP
  - o Heat Power   (kW)
  - o Annual Output   (kWh)
  - o Annual CO_2 Saving   (Kg)
  - o Electrical Power   (kW)
  - o Annual Output   (kWh)
  - o Annual CO_2 Saving   (Kg)
  - o Cost
- Monitoring System – (Manual / Remote / Real Time)

Appendix J: Contacts for Needs/Usability Studies

joe.baker@lbbd.gov.uk,
rachel.hogger@lbbd.gov.uk,
Michael.lai@barnet.gov.uk,
Ben.Thomas@bexley.gov.uk,
claire.smith@brent.gov.uk,
Jeff.Bartley@brent.gov.uk,
katie.ryde@bromley.gov.uk,
alastair.baillie@bromley.gov.uk,
ben.coombes@bromley.gov.uk,
indira.param@bromley.gov.uk,
Nicole.Fletcher@camden.gov.uk,
kirsteen.harrison@camden.gov.uk,
simon.mills@cityoflondon.gov.uk,
Emma.bara@cityoflondon.gov.uk,
Kia.Colbeck@croydon.gov.uk,
ian.weake@ealing.gov.uk,
kinghamc@ealing.gov.uk,
dobsonh@ealing.gov.uk,
Daisy.Johnson@enfield.gov.uk,
pippa.hack@greenwich.gov.uk,
barry.jenkinson@greenwich.gov.uk,
Jacquie.Goacher@hackney.gov.uk,
Andrea.Cronin2@hackney.gov.uk,
Jason.Davis@hackney.gov.uk,
stephen.mcandrews@lbhf.gov.uk,
sule.nisancioglu@haringey.gov.uk,
jill.warren@haringey.gov.uk,
Deborah.ganley@harrow.gov.uk,
rebecca.binns@harrow.gov.uk,
YuYao.Gong@harrow.gov.uk,
mark.lowers@havering.gov.uk,
Sheri.Lim@havering.gov.uk,
peter.williams@havering.gov.uk,
Rebecca.Davey@havering.gov.uk,
kwebster@hillingdon.gov.uk,
hlane@hillingdon.gov.uk,
Lucy.griffiths@hounslow.gov.uk,
ajit.matharu@hounslow.gov.uk,
rob.gibson@hounslow.gov.uk,
maxine.holdsworth@islington.gov.uk,
miranda.pennington@islington.gov.uk,
Caroline.Scott@islington.gov.uk,
laura.hales@islington.gov.uk,
Imogen.Purchas@islington.gov.uk,
joan.mcgarvey@rbkc.gov.uk,
krishna.desai@rbkc.gov.uk,
Frances.Smith@rbk.kingston.gov.uk,
jcurrie@lambeth.gov.uk,
jlissimore@lambeth.gov.uk,
iboulton@lambeth.gov.uk,
CLipop@lambeth.gov.uk,

Colin.Sandiford@lewisham.gov.uk,
Heather.Watkinson@lewisham.gov.uk,
Mike.Carless@merton.gov.uk
robin.whitehouse@newham.gov.uk,
bevan.jones@newham.gov.uk,
paul.umfreville@redbridge.gov.uk,
michael.doust@richmond.gov.uk,
sara.kassam@richmond.gov.uk,
l.broome@richmond.gov.uk,
Naomi.Baker@southwark.gov.uk,
bob.fiddik@southwark.gov.uk,
katherine.hudson@sutton.gov.uk,
katharina.winbeck@towerhamlets.gov.uk,
Rachel.Carless@towerhamlets.gov.uk,
brian.whiteley@walthamforest.gov.uk,
Carolyn.Seymour@walthamforest.gov.uk,
amcdonnell@wandsworth.gov.uk,
bgunn@westminster.gov.uk,
rwilson@westminster.gov.uk,
mleroy@westminster.gov.uk,
sagius@westminster.gov.uk,
Jared.Boow@londoncouncils.gov.uk,
agathe.m@gle.co.uk,
niall.machin@london.gov.uk,
nusrat.yousuf@london.gov.uk,
abigail.burridge@london.gov.uk,
Max.Dixon@london.gov.uk,
Jayne.Whitton@london.gov.uk,
will.lochhead@gol.gsi.gov.uk,
emma.bulmer@gol.gsi.gov.uk,
Verity.Ham@gol.gsi.gov.uk,
pbramwell.gol@go-regions.gsi.gov.uk,
Hannah.EDWARDS@gol.gsi.gov.uk,
HelenWoolston@tfl.gov.uk,
simonne.baker@environment-agency.gov.uk,
esther.vanlith@environment-agency.gov.uk,
steve.waller@idea.gov.uk,
gulnarhasnain@lda.gov.uk,
Richardjackson@lda.gov.uk,
lucy.poole@met.police.uk,
emma.devenish@met.police.uk,
Robert.Sambrooks@learningtrust.co.uk,
Christopher@lovelondon.org.uk,
g.burgess@lsx.org.uk,
s.jayasinghe@lsx.org.uk,
jonathan.turner@cslondon.org,
Andy.Deacon@london.gov.uk
Matthew.Carpen@london.gov.uk
thomas.carpen@london.gov.uk
suzanne.lemiere@london.gov.uk
Susanna.STRANDELL@gol.gsi.gov.uk

Appendix K: Needs Study Email

Adrian Hewitt's Email

Local Authority planning departments have (since 2005) had a statutory monitoring target of reporting on the generation of renewable energy in their area. With the adoption of renewable energy policies like the "Merton Rule" and the Mayor's policies, the amount of renewables is rising exponentially without any coherent way of keeping track of it.

In response to this, I am working with a university in Massachusetts (Worcester Polytechnic Institute) on building a simple but sophisticated web based system for tagging new buildings onto a Geographical Information System (electronic map), creating "Birth Certificates" for these new buildings, and then installing monitoring devices to them to record the energy performance of the building and the output of the renewable energy.

Attached is an excel spreadsheet of things that we want to ask developers when they submit an electronic planning application.

Can you scan through this and see if there is anything that we have forgotten and which would be useful to ask for - or if you have any general or specific comments.

We have been discussing this project with GOL and others, with the aim of creating a London wide computer system.

Team One's Email

We are a team of students from Worcester Polytechnic Institute in Worcester Massachusetts, USA. We are working for the next 5 weeks with the Planning and Projects Department of the Merton Council. The project we are assisting with is a system to monitor renewable energy. We have updated a table used by Adrian Hewitt to gather the necessary onsite energy information of buildings. If you could please briefly review this table for any possible errors or suggestions we would be very grateful. Thank you for your time and for your cooperation.

Appendix L: Usability Study Email

We are a group of students from the Worcester Polytechnic Institute of Worcester, Massachusetts, USA. We are currently working with Adrian Hewitt of the Planning and Projects Department in the Merton Council to develop an automated system for reporting energy usage in each building throughout the borough. This information will then be analyzed to determine whether the development is adhering to the Merton Rule. To obtain the initial information for new buildings, developers will be required to fill out a Renewable Energy Localized Installation Entry Form, RELIEF for short. This form asks for information such as building size, number of windows, and onsite renewable energy sources. The RELIEF then calculates the predicted CO_2 and displays the 10% that must be produced by the renewable energy sources. Attached is the RELIEF prototype. If you could please fill out this form as a developer, and include any questions or comments in the comments text box at the end we would be most appreciative. Please keep in mind that this is still in the developmental stages and will have a more intuitive user interface when finished.

Appendix M: MrHomer Software Architecture Document

Introduction

MrHomer is a framework designed and implemented by the C-2008 Merton IQP Team. The Team intends this document to be used by future developers (specifically, future IQP teams) for expansion and use. It will provide insight as to the system's design and proper use, as well as provide rationales for many of the design decisions made by the system's developers. MrHomer is a C# development. C# was selected as the primary language of choice because it allows for rapid development; because MrHomer is being used to develop a prototype and the final product is on a very strict and inflexible timetable, rapid development was an important consideration. In addition to C#, Visual Basic .NET may also be easily used to use and expand upon MrHomer. However, no tutorials or guides will be provided for Visual Basic .NET.

Purpose

The purpose of MrHomer is to provide a solid base from which all future code expansion can be performed without needing to rewrite large portions of code. The result is a code base that enables users to implement almost any feature needed without needing to alter or recompile the framework.

Definitions and Acronyms

- MrHomer: Merton Rule Hierarchical Online Monitoring and Energy Reporting
- IQP: Interactive Qualifying Project
- WPI: Worcester Polytechnic Institute
- RUP: Rational Unified Process
- Package: A group of software components designed to work together
- Class: A blueprint for a software structure consisting of behavior and data
- Object: The actual data and behavior constructed from a class
- UML: Unified Modeling Language

Architectural Representation

The architecture of MrHomer will be represented in the 4+1 Views' perspectives as recommended by the RUP. The 4+1 Views model includes a logical perspective, a development perspective, a process perspective, a physical perspective, and a scenario perspective. Each perspective is described in detail below, and each perspective's application to MrHomer is provided a section later in the document.

Logical View

The logical view provides an overview of the packages that the system provides. In addition, brief details will be given about each package, as well as concrete examples of when the package would be needed by the type of system MrHomer was designed to facilitate.

Development View

The development view provides a detailed overview of what a developer would work with. It expands upon each of the packages described in the logical view. In addition, tutorials will be provided for each visible class.

Scenario View

The scenario view will demonstrate various situations the framework's users might encounter, and a possible solution to them. It will demonstrate the key features of MrHomer.

Physical View

The physical view will provide an overview of how the MrHomer system integrates with a computer's existing hardware, as well as any considerations the system's users will need to take. MrHomer, at the time of writing, has no major hardware constraints, and as such, the Physical view will not be included in this document.

Process View

The process view will describe in detail the various components that may run concurrently. In addition, it will provide insight into deployment of MrHomer, its efficiency, as well as its potential scalability.

Logical View

MrHomer is divided into four packages, each of which contains all the pieces necessary to make it function at its most limited level of functionality. However, MrHomer is designed to run most effectively when used as a single framework. The four major packages are the Database package, the Security package, the Network package, and the Plug-in package.

Database

The Database package's entire purpose is to interface with query-based databases. The package, through its classes, will provide the ability to access databases, as well as modify them. A database could be used to store any amount of information the program needs, and as such, is an extremely useful portion of MrHomer. The Database package as a whole will also be available from within any portion of any program using the MrHomer framework, providing software engineers with a simplified system for data collection and storage.

Example

A programmer is writing a user management system. Rather than store all data in an undocumented format, the programmer can use MrHomer's Database system to store them. The programmer would create a table for users, and through MrHomer, access that table and modify its members.

Example

A programmer wishes to store readings from sensors. Rather than store data in program memory (a poor practice due to potential program crashes as a result of misfortune or other non-program-related circumstances) or in temporary files, the programmer can put the values them into a database.

Example

A programmer is writing a program that interfaces with a website hosted on a computer. Rather than modify HTML and use program data, the programmer can modify a MySQL or PostGres database, which the website can then use as its source for data.

Security

The Security package's purpose is to provide users with an accessible way to perform security-related functions throughout a program. It provides no actual security functionality on its own, but rather a suggested way of performing security. With it are interfaces whose implementation details are up to the user. However the user chooses to secure its data, the security package provides accessible means to it.

Example

A programmer is dealing with sensitive data, and needs to store it. Rather than write an unreliable program-specific way of encrypting it, or dumping it in an obscure location, the programmer can implement a few of the Security interfaces, and use the MrHomer security package in all programs that need that data, thus securing the data.

Example

A programmer needs to send data across the internet. Rather than send raw data which can then be viewed by unwanted eyes, as well as more easily tracked, the programmer can implement a few of the Security interfaces, and mangle the byte stream such that only the correct MrHomer program on the other end can decrypt it.

Network

The Network package's purpose is to provide users with an easy way to send data not only across a network, but also across the Internet. It removes the need for users to deal with the tangle of processes needed to send and receive data from a network. The package also provides utility classes and interfaces to allow data to be more easily sent across a network connection.

Example

A programmer needs to send data across the internet. The programmer can use the Network package to wrap that package into an easily sent format, and to send the data to its destination. The MrHomer network package on the receiving end will then get and interpret the data as needed.

Plug-In

The Plug-In package's purpose is to provide the ability for users to write code that the final program can use without knowing anything about it. In addition, several utility classes are provided to allow programmers to be selective about their plug-ins, and to enforce strict standards.

Example

A programmer knows the program will need to read information from sensors, but has no idea how the sensors would work. The programmer can use MrHomer's Plug-In system to establish a set of rules for how sensor plug-ins will give the program information, but not how the plug-in will do it.

Example

A programmer wants to allow a user to customize how their program behaves. The programmer can use MrHomer's Plug-In system and define what the behavior needs to accomplish. Users can then write miniature programs adhering to the defined rules, and MrHomer will allow the program to use the user's code.

Development View

MrHomer's packages are each divided into several pieces, each of them having several things in common. Each of them has a Manager, which will be the way to access most in-package data structures used in the course of the program. There are a few exceptions, as will be outlined below.

Database

The Database package is split into two classes, and an interface that need not be implemented by users, but the option is still available.

DatabaseManager

The database manager simplifies access to the currently connected databases. By centralizing the connection options, the user does not need to maintain multiple database objects and pass these to functions that require database access.

Code Example
```
DatabaseManager dm = DatabaseManager.getInstance();
```

IDatabase

Since many different database engines exist, a common way to access them is necessary. The IDatabase interface provides the standard template for a database. Classes implementing this interface are responsible for any necessary translation of queries to work on their engine.

Code Example
```
public class SQL_Database : IDatabase
{
  protected SqlConnection conn;

  public SQL_Database() { }

  public virtual bool Connect(string dataSource, string username,
string password, string database)
  {
    // Create a connection String
    conn = new SqlConnection(ConnectionString);
    conn.Open();
  }

  public void Disconnect()
  {
    conn.Close();
  }

  public DBResults Query(string queryString)
```

```
        {
            SqlCommand n = new SqlCommand(queryString, conn);
            SqlDataReader r = n.ExecuteReader();
            // Create a result set
            r.Close();
            return res;
        }

        public bool IsConnected()
        {
            return conn != null;
        }
    }
```

DBResults

Database queries return rows of data, which are normally accessed sequentially. Because this is very slow if a previous row is needed, the DBResults class encapsulates a table of results much like is displayed in a visual RDBMS.

Code Example
```
        DBResults r = new DBResults();
        r.AddRow(new string[] { "Key1", "Key2" }, new object[] { 1, 2 };
        Dictionary<string, object> firstRow = r[0];
```

Security

As this system is responsible for gathering data, storing it, relaying it across a network, and allowing users to view it, security is a necessity. The framework for the ERS includes a Security package for this purpose. This document explains the components of the Security package and recommended practices for implementing the Security component.

Security Manager

The core component of the Security package is the Security Manager. The Security Manager is responsible for storing the active Key Manager, Authenticator, and Cryptographer. These components are explained in detail later. The Security Manager acts as a Singleton, meaning that only one instance of the Security Manager will exist within the Application at any time. This allows trivial access to security features via the Security Manager.

Code Example
```
        SecurityManager sm = SecurityManager.getInstance();
        SecurityManager sm = SecurityManager.getInstance(Parameters);
```

ICryptographer

ICryptographers are responsible for encrypting and decrypting data. They are given a byte stream, and an array of other data they might need, and they encrypt or decrypt the byte stream.

Code Example

```
public class CryptoDemo : ICryptographer
{
  public byte[] Encrypt(byte[] data, object[] Details)
  {
    return data;
  }

  public byte[] Decrypt(byte[] data, object[] Details)
  {
    return data;
  }
}
```

IAuthenticator

IAuthenticator objects are responsible for verifying the identity of a client, server, or user. The specific implementation is left to future developers, but the groundwork is in place. If the Authenticator is to be used with the Security Manager, it must implement the IAuthenticator interface. The Security Manager implements this interface, so it can be used as an Authenticator if, for some reason, this is desired (for example, if a custom function expects an IAuthenticator).

Code Example

```
public class AuthDemo : IAuthenticator
{
  public bool IsRegistered(string Name, object[] Details)
  {
    return true;
  }

  public bool Authenticate(string Name, object[] Details)
  {
    return true;
  }
}
```

IKeyManager

Classes implementing IKeyManager handle either user or machine keys (or both). IKeyManagers are responsible for providing a caching mechanism if one is desired. IKeyManager provides methods to set or retrieve a key.

Code Example

```csharp
public class KeyDemo : IKeyManager
{
  public object[] GetKey(string Name, object[] Details)
  {
    return new object[] { new byte[] { 1 } };
  }

  public void SetKey(string Name, object[] Details) { }
}
```

Network

The Network package is divided into two classes and one interface. The only one the user absolutely needs to use is the Connection Manager, but the other two are helpful as well.

Connection Manager

The Connection Manager is the class through which the connection package is accessed. Through this class, the user will be able to connect to other MrHomer-based programs, listen for connections from other MrHomer-based programs, and once connections are established, send and receive data. More detailed code examples are provided in the Scenario view section.

Code Example

```csharp
ConnectionManager cm = ConnectionManager.getInstance();
```

IHSerializable

IHSerializable is an interface that must be implemented by the user in any number of ways the user wants. It provides a standard way of communicating with the Packet class, to allow any data types to be encapsulated as a byte stream. If the user plans on using any classes across networks, they are strongly advised to implement this interface.

Code Example

```csharp
//Class StringData is an IHSerializable container for strings.
public class StringData : IHSerializable
{
  private string data;
  public string Data
  {
    get { return data; }
    set { data = value; }
  }

  public byte[] Serialize()
  {
    System.Text.UTF8Encoding encoder = new UTF8Encoding();
    return encoder.GetBytes(data);
  }
```

```
      public void Deserialize(byte[] myData)
      {
        System.Text.UTF8Encoding encoder = new UTF8Encoding();
        data = encoder.GetString(myData);
      }
    }
```

Packet

The Packet class provides a standard way for users to send and receive data. It has fields much like an email, where a user can specify a sender, a recipient, and subject, and an IHSerializable body. The sender, recipient, and subject are all strings.

Code Example
```
    //If a user wants to encapsulate a string, using the StringData class
    //we created earlier.
    StringData d = new StringData();
    d.Data = "This is a test!";
    Packet p = new Packet("Sender", "Recipient", "Test", d));
```

Plug-in

The Plug-in package provides three components, one class the user must use, one class the user may choose to use, and one interface that must also be used.

Plug-in Manager

The Plug-in Manager acts as a central hub for all Plug-in-Related activity. Obtaining a reference to the Plug-In Manager is simple as it implements Singleton functionality to ensure there is one Plug-in Manager within a program. Specific examples for Plug-in Manager functionality will be provided in the Scenario view.

Code Example
```
    //If a user wants the PluginManager reference
    PluginManager pm = PluginManager.getInstance();
```

IPlugin

IPlugin acts as an indicator that an interface should be loaded as a plug-in type. If this interface is not implemented, the Plug-in Manager will ignore a plug-in blueprint in the form of an interface.

Code Example
```
    //If a user wants a sensor plugin blueprint, it might look something
like
    //this.
    public interface ISensor : IPlugin
    {
      double RequestData();
```

```
        string GetUnits();
    }
```

MrHomerPluginAttribute

The MrHomerPluginAttribute class allows a user to specify information about his/her plug-in. In addition, if a programmer wishes that all plug-ins adhere to a specific standard of information.

Code Example

```
    //If a user wishes that "Name" be a standard property, a plug-in
developer
    //would be required to insert the following above the class definition
    [MrHomerPlugin("Name", "My Super-Awesome Plug-In")]
    public class MyPlugin : IMyPlugin
```

Scenario View

Provided below are detailed examples of how to accomplish specific goals within each package, as well as some cross-package examples. This list is not exhaustive; MrHomer can be used in other ways. Refer to the API for details.

Database

Cross-Class Examples

Code Example

```
    // Getting a database manager
    DatabaseManager dm = DatabaseManager.getInstance();

    //Connecting to a database
    dm.ConnectDatabase("Name", "DB_Type", "dbserver:1234", "User",
"Password", "MyDB", null);

    // Running a query
    DBResults r = dm.QueryDatabase("Name", "SELECT * FROM SampleTable WHERE
Condition = True;");

    // Using the results
    foreach(Dictionary<string, object> row in r)
      Console.WriteLine("Row: " + r.ToString());

    // Closing a connection
    dm.DisconnectDatabase("Name");
```

Security

Cross-Class Examples

Code Example

```
    // Getting a Security Manager
```

Study of City Knowledge in Merton

February 21, 2008

```
SecurityManager sm = SecurityManager.getInstance(Auth, Crypto, KeyMan);

// Authenticating a User
sm.Authenticate("User", null);

// Finding a Host Key
object[] key = sm.GetKey("Host", null);

// Encrypting a Packet
byte[] packet = sm.Encrypt(message, key);

// Decrypting a Packet
byte[] data = sm.Decrypt(message, key);
```

Network

ConnectionManager

Code Example

```
//Getting a connection manager
ConnectionManager cm = ConnectionManager.getInstance();
```

Code Example

```
//Required initialization. 3 method calls.
//Set the credentials first. It's like your system's
//identifier, or signature.

//Set the request packet second. It must be equal on
//any two systems communicating. It ensures valid
//programs are communicating.

//Set the dropout. Connections that don't receive
//credentials will automatically drop after the
//timeout if it's true. Otherwise, they will attempt
//to get credentials forever. It is recommended to
//set it to true.

//Also, attempted connections to unresolved host
//names will drop after the timeout, no matter what.
byte[] credentials = new byte[0];
  //Set to something meaningful
byte[] request = new byte[0];
  //Set to something meaningful
uint timeout = 10000;
bool drop = true;
cm.SetCredentials(credentials);
cm.SetRequestPacket(request);
cm.SetDropout(timeout, drop);
```

Code Example

```
//Listening on a port
//Note, the second parameter must be "true" for now.
//cm.ListenForConnection(port number, using IPV4);

//To stop listening, call StopListening.
cm.ListenForConnection(1337, true);
cm.StopListening(1337);
```

Code Example

```
//Connecting to a remote lost
//Note, last parameter must be "true" for now.
//cm.MakeConnection(Connection Label, Host, Port, using IPV4);
cm.MakeConnection("MyConnection", "192.168.200.35", 65535, true);
```

Code Example

```
//Getting information about the current state.

//Active connections are successful connections you initiate.

//Inactive connections are successful connections made on listening ports.
//Note that inactive connections also return a the connecting computer's
//credential packet that was set earlier.

//Failed connections are connections you initiate that failed to connect.
//Dropped connections are formerly active or inactive that dropped without
//user action. They both require a parameter. True to delete the data
//after its returned, false to retain it for future reference.
cm.GetAllActiveConnections();
cm.GetAllInactiveConnections();
cm.GetDroppedConnections(false);
cm.GetFailedConnections(false);
cm.GetListeningPorts();
```

Code Example

```
//Activating an inactive connection can be done like so.
string oldname = "unknown0";
string newname = "MyOtherConnection";
cm.ActivateConnection(oldname, newname);
```

Code Example

```
//To send data, call SendData.
string connection = "MyConnection";
byte[] data = new byte[0];
  //Set to something meaningful.
```

```
        cm.SendData(connection, data);
```

Code Example

```
        //To receive data, call GetData. It has two overloads.
        //One accepts a connection name and a clear flag. The
        //other accepts only a clear flag. Set to true, the clear
        //flag will clear all data that was returned from the cache.
        //Without a name, it returns all data on all connections that
        //has been received. With a name, it returns only data that
        //connection has sent.

        string rConnection = "MyOtherConnecton";
        bool ClearFlag = false;
        cm.GetData(rConnection, ClearFlag);
        cm.GetData(ClearFlag);
```

Code Example

```
        //To terminate an active connection, call KillActiveConnection.
        //To terminate an inactive connection, call KillInactiveConnection.
        string active = "MyConnection";
        string inactive = "unknown0";
        cm.KillActiveConnection(active);
        cm.KillInactiveConnection(inactive);
```

Packet

Code Example

```
        //Creating a new packet from your own data is as simple as
        //calling the constructor with information.
        string sender = "Me!";
        string receiver = "You!";
        string subject = "Hi!";
        IHSerializable body = new StringData("How are you?");
        Packet p = new Packet(sender, receiver, subject, body);
```

Code Example

```
        //Turning a packet into a byte stream is as easy as calling
        //Serialize.

        byte[] bytes = p.Serialize();
```

Code Example

```
        //Turning a byte stream into a packet can be done by
        //creating a new packet object.

        Packet newPacket = new Packet(bytes);
```

Cross-Class Examples

Code Example

```
//The Packet class can be used to create the byte streams
//needed by ConnectionManager.

ConnectionManager cm = ConnectionManager.getInstance();
Packet credentials = new Packet("Chris",
        "You",
        "My Credentials",
        new StringData("Let me show you them"));
Packet request = new Packet("Me",
        "You",
        "Request",
        new StringData("Plz?"));
cm.SetCredentials(credentials.Serialize());
cm.SetRequestPacket(request.Serialize());
```

Code Example

```
//It can also be used to send information, and to debug it
//when it is received.

Packet message = new Packet("Brian",
        "Bob",
        "Text",
        new StringData("S'up."));
cm.SendData("BobConnection", message.Serialize());

List<byte[]> data = cm.GetData("BobConnection", false);
Packet incoming = new Packet(data[0]);
```

Plug-in

Plug-in Manager

Code Example

```
//Getting the PluginManager is as easy as calling its
//Singleton method.
PluginManager pm = PluginManager.getInstance();
```

Code Example

```
//To load plugin types, call LoadInterfaceTypes.
pm.LoadInterfaceTypes("myDLL.dll");
```

Code Example

```
      //To load actual plugins once types are loaded, call LoadPluginTypes.
      pm.LoadPluginTypes("myOtherDLL.dll");
```

Code Example
```
      //To get a list of all plugin types loaded, call getInterfaces.
      pm.getInterfaces();
```

Code Example
```
      //To get a set of plugins given a type, call getPlugins.
      Type IPluginType = typeof(ISensor);
      List<Type> plugins = pm.getPlugins(IPluginType);
```

Code Example
```
      //To create an instance of a plugin once you have its type,
      //use the Assembly class.
      Type plugintype = plugins[0];
      Assembly asm = Assembly.GetExecutingAssembly();
      ISensor myPlugin;
      myPlugin = asm.CreateInstance(plugintype.FullName)as ISensor;
```

Process View

There are a few notes that need to be made about MrHomer, specifically with respect to Threading, Deployment, and Scalability.

Threading

MrHomer, on the surface, is procedural. That is, MrHomer can be used by anybody who knows how to write simple, one-process code. However, underneath the simple API lies a multi-threaded system with constant update mechanisms. MrHomer's internals are entirely thread-safe. The developers recommend several MrHomer functions, such as networking, be running on their own threads. However, it is not necessary.

Deployment

MrHomer has several requirements. Simply, MrHomer requires the .NET Framework 2.0 from Microsoft to be used. In addition, programmers using MrHomer need two things.

- A .NET language compiler, such as Video Studio .NET, or Visual C# Express
- MrHomer.dll

Scalability

MrHomer's individual components are highly scalable. Given an increased workload, they will function just as expected, with no major differences in processing time. As such, MrHomer can be used for any project, however large or small.

Appendix N: Security Protocol

Authentication

The following procedure describes the basis for system initialization.

Assumptions

- Alice is online.
- Bob knows how to locate Alice on the network.
- For any public key Q', there is a corresponding private key P', and vice-versa.
- The private key is known only to the system it is issued for.
- A system has the public key of any systems allowed to communicate with it.

Procedure

1. Bob is powered on. Bob is one of Alice's clients.
2. Bob sends a connect request to Alice. This request is encrypted using Qa.
3. Alice receives a request from what appears to be an unknown client.
4. Alice decrypts the message with Qa, and discovers that the request is from Bob.
5. Alice forms a response, and signs it with Pa.
6. Alice encrypts the response with Kb and sends it to Bob.
7. Bob receives the response, and decrypts it with Qb.
8. Bob assumes that the message is from Alice, and uses Ka to verify the signature.
9. Bob verifies the signature. (If verification fails, the connection is aborted)
10. Bob generates a new key Dab and sends it to Alice signed with Pb (possibly encrypted with Ka).
11. Alice receives the response, verifies the signature, stores key Dab, and adds Bob to the list of connected clients.

At this point, both Alice and Bob have confirmed the identity of the other system, and have established an open communication channel. They have also agreed on a key to be used only for this session for encrypting data (this is faster than using two keys and protects key security). If at any time Alice fears this session key has been compromised (or on a timer), she may revoke the key and issue Bob a new one. If Bob fears the key has been compromised, he may request that Alice issue a new one.

Chaining

The following procedure describes the process by which a system above Alice may talk to Bob.

Background

Suppose that Charlie is one level above Alice, and that Charlie and Alice have completed authentication. Charlie would like some data from Bob. Charlie knows of Bob's existence through Alice, but cannot talk directly to Bob. To conserve bandwidth, Charlie wishes to talk directly to Bob.

Assumptions

- Charlie does not have Qb.
- Bob does not have Qc
- Both Bob and Charlie have authenticated with Alice.
- All data transfers are encrypted with the appropriate key for transfer between the two systems.

Procedure

1. Charlie sends a message to Alice requesting a direct link to Bob.
2. Alice verifies Charlie's permission to do so. (If verification fails, abort)
3. Alice sends a message to Charlie containing Qb.
4. Charlie acknowledges receipt of this message, and awaits a connection from Bob.
5. Alice sends a message to Bob containing Qc and the address of Charlie.
6. Bob initiates the authentication procedure with Charlie.

Upon completion of the authentication procedure, Bob and Charlie are connected, and Charlie is able to access the same data from Bob as Alice.

Outside Users

The following procedure describes the process for allowing outside users to access the system.

Background

Debbie is a user who works for Alice's owner. Debbie needs to access the information from Bob, but is currently away from Alice. Debbie has some software on her computer to connect to Alice, and wants to use this to view the data Alice is recording.

Assumptions

- Prior to the remote connection
    - Debbie was issued Pd (which could be a password).
    - Qd was recorded into Alice's list of remote users.
    - Debbie's program was configured with Ka, but Debbie needs not be aware of this.

Procedure

1. Debbie starts her client, which connects to Alice. The connection request is encrypted with Ka.
2. Alice receives the request, decrypts it, and sees that Debbie is a remote user.
3. Alice responds to Debbie with a message encrypted with Qd and signed with Pa.
4. Debbie's client verifies the signature, and requests Pd from Debbie, the decrypts the message with Pd.
5. Debbie's client computes a temporary key, Dad, and signs this with Pd (optionally encrypted with Qa), which is then sent to Alice.
6. Alice verifies the signature, and adds Alice to the connected remote users list.

At this point, Debbie is connected to Alice, and both Alice and Debbie's client have confirmed the other's identity. This process is identical to authentication, excluding the prompt for the private key and the difference in the final step. The remote users list would enable the system to provide limited access to other systems. In addition, the Outside Chain Access protocol below would be enforced. Now, Debbie's client would display the same information that was available to Charlie.

Outside Chain Access

The following details how a remote user, once connected, can access systems below the connected host.

Background

Once a Debbie has connected to Alice, she would like to view information from Bob. Because she works for Alice, she has authorization to access this data.

Assumptions

- Alice and Debbie are authenticated.
- All data transfers are encrypted with the appropriate key for transfer between the two systems.

Procedure

1. Debbie sends a message to Alice requesting information from Bob.
2. Alice knows that Debbie is a remote user, and confirms that Debbie has authorization.
3. Alice re-encrypts Debbie's message with Dab and forwards it to Bob.
4. Bob receives the request from Alice, but has no idea it was originally from Debbie.
5. Bob responds to Alice's request with the appropriate data.
6. Alice decrypts the message, then re-encrypts it with Dad and sends it back to Debbie.

In this way, Alice would act as a proxy for Debbie, protecting the system from rogue clients. It also eliminates the need for each system to have the same user lists. The overall bandwidth increase for this feature would be minimal, as the Chaining protocol would still be used.

Data stored for the above protocols

This section outlines the storage needs for the system described above.

Alice

Alice would need to know about systems, which report to her, systems to which she reports, and remote users who work for her. Therefore, Alice needs this information:

- Other Systems
    - The public key
    - The identifier for the system (e.g. the name)
    - If this system is above (Charlie) or below (Bob)
    - The current session key for this system (only if connected)
    - The session key validity timeout (if desired)
- Remote Users
    - The public key
    - The identifier for the system (e.g. the name)
    - The current session key for this client (only if connected)
    - The session key validity timeout (if desired)
    - A flag indicating permission to access the chain
- Private Key

Bob

In the overall system, Bob and Alice are equivalent. Therefore, the same data structure would be required. Data for different systems would be stored.

- Other Systems
- Remote Users
- Private Key

Charlie

As with Alice and Bob, Charlie is equivalent in the overall system

- Other Systems
- Remote Users
- Private Key

Debbie

Debbie is the only system in this overview that is unique. Debbie needs only to store her private key and Alice's public key.

- Public key of remote server
- Private Key

Appendix O: Security Guide

Abstract

This document outlines the proper method for implementing security in the Energy Reporting System (ERS). It is primarily a technical document, intended for implementers and not for general consumption. This document is informative; it provides guidelines, but does not require an exact implementation. The designers advise the reader to follow these guidelines. These guidelines are based on the current development of ERS and are subject to change.

Security Overview

As this system is responsible for gathering data, storing it, relaying it across a network, and allowing users to view it, security is a necessity. The framework for the ERS includes a Security package for this purpose. This document explains the components of the Security package and recommended practices for implementing the Security component.

The Security Manager

The core component of the Security package is the Security Manager. The Security Manager is responsible for storing the active Key Manager, Authenticator, and Cryptographer. These components are explained in detail later. The Security Manager acts as a Singleton, meaning that only one instance of the Security Manager will exist within the Application at any time. This allows trivial access to security features via the Security Manager. The Security Manager provides six methods to the application:

- Authenticator Methods

    o Is Registered – Uses the Authenticator to determine if a client, server, or user is registered.

    o Authenticate – Uses the Authenticator to verify the identity of a client, server, or user.

- Cryptographer Methods

    o Encrypt – Uses the Cryptographer to encrypt a message

    o Decrypt – Uses the Cryptographer to decrypt a message

- Key Manager Methods

    o Get Key – Retrieves a key from the key store

    o Set Key – Changes a key stored in the key store

These methods are explained in their respective sections below.

Authenticator

The Authenticator is responsible for verifying the identity of a client, server, or user. The specific implementation is left to future developers, but the groundwork is in place. If the Authenticator is to be used with the Security Manager, it must implement the Authenticator interface. This interface provides the following methods:

- `bool IsRegistered(string Name, object[] Details)`

    o Checks to see if the specified client, server, or user is registered

    o Name: The name of the client, server, or user

    o Details: Any additional data that is passed to the Authenticator. This can include, for example, a timestamp or a network connection

- `bool Authenticate(string Name, object[] Details)`

    o Verifies the identity of a client, server, or user

    o Name: The name of the client, server, or user

    o Details: Any additional data that is passed to the Authenticator. This can include, for example, a timestamp or a network connection

The Security Manager implements this interface, so it can be used as an Authenticator if, for some reason, this is desired (for example, if a custom function expects an Authenticator).

Multi-Authenticator

The Multi-Authenticator is a special form of Authenticator. It is configured initially with a list of acceptable Authenticators. When either of the Authenticator functions is invoked, the Multi-Authenticator will test each of the Authenticators in its list, until either the list is exhausted

or a match is found. The Multi-Authenticator implements the Authenticator interface, so its use is transparent to the Security Manager.

Cryptographer

Any data transmitted over a network in plain text is subject to sniffing attacks. Worse, if an attacker analyzes the packet structure, he can construct his own packets and inject them into the network stream. Encrypting the data will not eliminate this threat, but significantly lessens its impact. To facilitate the encryption of data, the Cryptographer interface is presented. It supports only the two methods below:

- `byte[] Encrypt(byte[] data, object[] Details)`

    o Takes a byte stream, encrypts it, and returns the encrypted version

    o Data – the byte stream to encrypt

    o Details – any additional information the Cryptographer needs, including encryption keys.

- `byte[] Decrypt(byte[] data, object[] Details)`

    o Takes a byte stream, decrypts it, and returns the encrypted version

    o Data – the byte stream to decrypt

    o Details – any additional information the Cryptographer needs, including encryption keys.

These functions must be complements of each other, such that a byte stream *b* satisfies the following requirements:

- `b = Decrypt(Encrypt(b));`

- `b = Encrypt(Decrypt(b));`

    `Again, the Security Manager implements this interface.`

Multi-Cryptographer

The Multi-Cryptographer allows multiple stages of encryption to be used. Cryptographers are added to the Multi-Cryptographer in advance. The Multi-Cryptographer implements the

Cryptographer interface, so it can be transparently used in place of a single Cryptographer. When the Multi-Cryptographer's Encrypt function is invoked, each of the configured Cryptographers encrypts the data output by the preceding Cryptographer. Decrypt works in the opposite fashion: the Cryptographers are used in the reverse order.

Key Manager

The key manager is responsible for providing lookup for user, client, and host keys. The system can be use for any purpose, from authentication to encryption. The key manager provides two functions of interest:

- `object[] GetKey(string Name, object[] Details)`

  o Looks up the key for the named user, client, or server

  o Name – the object for which the key is requested

  o Details – any additional information the Key Manager needs

- `void SetKey(string Name, object[] Details)`

  o Changes the stored key for the named user, client, or server

  o Name – the object for which the key is requested

  o Details – any additional information the Key Manager needs

The Security Manager implements this interface as well.

Multi-Key Manager

In many cases, it is desirable for keys to be stored in different systems. As an example, consider the Security Manager running on a Windows system. Keys for servers could be stored in the Kerberos engine, while client keys are stored in an SQL database. Supporting multiple Key Managers allows this type of key storage without requiring a specific implementation.

An important note is the method the Multi-Key Manager uses to change stored keys. Only the first Key Manager in the list receives the new key. Although this behavior seems unexpected, it is the only feasible approach to updating a key. Determining where a key should be stored, if it is not stored already, requires immensely complex and application-specific logic.

A consequence of the key storage mechanism is the use of key caching. If the key cache were set as the first element in the Multi-Key Manager, then this behavior would update the cache. The Multi-Key Managers checks the Key Managers in increasing order for key lookup, so the cache would serve future requests for this key.

Custom Security Features

By defining these interfaces, the development of custom security components is trivial. It is not necessary to create all three types of security classes; the application provides several defaults. However, there are two key points to consider. First, the default classes do not provide maximum security or efficiency. Second, each system wishing to communicate using a new Cryptographer must have the binary installed. It is left to the developer to resolve these two issues.

Appendix P: Plug-In Guide

Introduction

In this document, Plug-Ins will be explained in full detail, leaving no detail untouched. Before you begin, it is recommended that the MrHomer Software Architecture Document also be read, as it will provide more insight into the system's design and use.

What are Plug-Ins?

Plug-ins are small programs that interface with a larger system. The larger system, however, does not need the plug-in to function properly. A plug-in is essentially the software version of a USB device, while the larger system is the computer, and the interfacing API is the USB port. Plug-ins can be used to add a variety of content to a program, or to add additional functionality.

Why would I want Plug-Ins?

Plug-ins are a versatile way to allow users to add additional functionality to programs. They allow new features to be implemented without needing to change any of the base source code. In addition, they allow the developer to add a slew of features to a program quickly and easily. This allows main program engines to be small, clean, easily tested, and it reduces the need to rewrite or recompile programs to add more functionality.

Example

Adobe Photoshop® uses plug-ins for many of its internal tasks. File formats (such as JPEG, Bitmap, GIF, and PNG) are all opened through plug-ins for each respective file type. Filters are also plug-ins, and user-created filters are common all over the Internet, as well as guides for writing custom filters.

Example

Web Browsers all use plug-in architectures to allow users to access different types of media. For example, Adobe Flash® SWF files are opened through the Flash Player or Shockwave plug-ins. In addition, several media players (such as QuickTime, Windows Media Player, and RealPlayer) include plug-ins in their default installation packages to allow audio and video files to be played using their plug-in on a web site.

Example

Media players all use plug-in capabilities to allow multiple file formats to be played. File formats such as MP4, MP3, AVI, and WMV, among others, are all opened through a video decompression algorithm, which is a standardized plug-in all media players are designed

around.

Writing a Plug-in System

While writing a plug-in system is not generally difficult, many issues can arise. MrHomer seeks to eliminate all the possible troubles with Reflection, storage, and validation by providing a simple, easy-to-use API for writing a plug-in system.

Before You Start

Read the MrHomer Software Architecture Document to get a good grasp of how the system is generally designed and used. In addition, plan what plug-ins are needed, and plan a way for a user to load plug-ins. Will they be in a folder? Will they be loaded by a menu option or button? Forethought is key to successful design, and as such, is highly recommended.

Example

Adobe Photoshop® stores all of its plug-ins in the "Plug-Ins" folder in the program director. They are automatically loaded when the program starts, thus masking the actual plug-in process from the user. Other programs, like Bethesda Softworks' Oblivion allow users to select plug-ins via an in-game user interface.

System Design

At this stage, needed plug-ins should be planned out, as well as how plug-ins will be loaded. However plug-ins are planned to be used, make certain that your subsystem will be able to handle as many plug-ins as possible, or in the event that only one will be used, will be able to switch them out intelligently.

Defining Plug-in Types

At this point, produce an interface for each plug-in type. Each plug-in interface must inherit from ISensor found in the MrHomer.Plugin package; otherwise, MrHomer's Plug-in Manager will ignore them.

Example

A programmer is writing a system that requires sensors. The programmer, however, does not know what sort of sensors will be used or how to accept data from them. He decides the best course of action would be to make sensors plug-ins. A plug-in interface similar to below may be implemented:

```
public interface ISensor : IPlugin
{
  double RequestData();
  string GetUnits();
  void SetName(string name);
  byte[] SerializeConfiguration();
  void DeserializeConfiguration(byte[] data);
```

```
      }
```

The programmer also knows that the data must be interpreted, but he does not know which way people will want the data. He decides to include another plug-in for interpreting the data.

```csharp
  public interface IDataFilter : IPlugin
  {
    void Filter(DataReading data);
    byte[] SerializeConfiguration();
    void DeserializeConfiguration(byte[] data);
  }
```
At this point, the programmer has two plug-in types defined.

There are two options. Either the plug-in interfaces are put in a Dynamic Link Library (DLL) somewhere (recommended, as people will have more difficulty extending types if they cannot include the DLL in their code), or have it compile as part of the main executable (EXE) program. MrHomer allows both.

Loading Plug-in Types

In order to load the interfaces as valid plug-in types, the Plug-in Manager needs to be used. The Plug-in Manager provides two methods for loading plug-in interfaces.

Example

The first method for loading plug-in interfaces involves using a string to a dynamic link library.

```csharp
PluginManager.getInstance().LoadInterfaceTypes("MyDLL.dll");
```

The second method for loading plug-in interfaces involves passing in an Assembly. However, it is simple. To use the current Assembly (note, a project is an assembly):

```csharp
PluginManager pm = PluginManager.getInstance();
pm.LoadInterfaceTypes(Assembly.GetExecutingAssembly());
```

Both methods will work. However, as stated earlier, using DLLs is recommended.

Loading Plug-ins

Once your plug-in types have been loaded into the plug-in manager, plug-ins may be loaded. Without loading the appropriate plug-in types first, MrHomer will ignore the plug-ins! Plug-ins are loaded in much the same fashion as interfaces. There are two methods.

Example

The first method for loading plug-ins involves using a file path to a dynamic link library.

```
PluginManager.getInstance().LoadPluginTypes("MyDLL.dll");
```

The second method for loading plug-ins involves passing in an Assembly. However, it is simple. To use the current Assembly (note, a project is an assembly), use the Assembly static methods. However, this method will only work to load plug-ins that were compiled with the rest of the program.

```
PluginManager pm = PluginManager.getInstance();
pm.LoadPluginTypes(Assembly.GetExecutingAssembly());
```

Both methods will work. However, as stated earlier, using DLLs is recommended.
MrHomer is now configured to use plug-ins of your design! Finding the appropriate path to the DLL, however, is not included in the system, as it is application-specific.

Enforcing Policies

MrHomer includes a custom attribute called the MrHomerPlugin Attribute. Essentially, both the key and value are strings in a key-value attribute. They can be used to enforce certain levels of metadata be implemented by the plug-in.

Example

Using the ISensor example from earlier, suppose the plug-in writer wants to make sure the user defines a name for the plug-in, along with a description, a make, and a model of the sensor the plug-in is affecting. While it could be included in code, why not use metadata to force the data? ISensor can thus be modified to:

```
[MrHomerPlugin("Required", "Name")]
[MrHomerPlugin("Required", "Make")]
[MrHomerPlugin("Required", "Model")]
[MrHomerPlugin("Required", "Description")]
public interface ISensor : IPlugin
{
   double RequestData();
   string GetUnits();
   byte[] SerializeConfiguration();
   void DeserializeConfiguration(byte[] data);
}
```

Using the Plug-in Manager, enforcement of plug-in metadata is now possible. The first step is to obtain all the plug-ins that use the ISensor interface.

```
List<Type> plugins = pm.getPlugins(typeof(ISensor));
```

Once all the types have been obtained, they can be pumped through the enforcement method.

```
pm.EnforceAttributes("Required", myType);
```

It will return true if all the attributes in ISensor marked "Required" also appear in the plug-in. This can be used to enforce multiple types of metadata. For example, "Required,"

"Optional1," and "Optional2" can be used to enforce groups of metadata, but not all of it. An example plug-in for ISensor appears below with proper metadata.

```
[MrHomerPlugin("Name", "Software Sensor: Random")]
[MrHomerPlugin("Make", "Fake Sensor")]
[MrHomerPlugin("Model", "Wun-2-fif-T")]
[MrHomerPlugin("Description", "I am a description. Rawr!")]
public class RandomSensor : ISensor
{ }
```

All the attribute values may be obtained through MrHomer as well. Included is a method for pulling out attribute details.

```
List<MrHomerPluginAttribute> list = pm.getAttributes(myType);
```

Refer to the API for more details on MrHomerPluginAttribute.

Using Plug-In Objects

MrHomer includes two methods for instantiating plug-in objects. Both are detailed below. They are essentially the same method, however, one removes the need to cast the final object, and will simply set the object to null if there are any errors.

Example

The first and recommended method is the Generic method. The second parameter is simply an object array of parameters passed into the constructor. It is recommended that in documentation it is specified what constructor signatures should be for all plug-ins.

```
ISensor sensor = pm.CreateInstance<ISensor>(myType, null);
```

The second way is simply to cast the object from the non-generic type.

```
ISensor mysensor = (ISensor)pm.CreateInstance(myType, null);
```

Using the information presented above, a robust and easy-to-use application-specific system is very possible. It is recommended that in addition to this guide, programmers provide end users with plug-in programming guides of their own, as this document does not cover implementation-specific plug-ins.

Writing Plug-ins for a MrHomer Application

Writing plug-ins for a MrHomer application is very similar to implementing an interface for any other program. In addition to the interface, several attributes may need to be

implemented. The program should contain additional documentation to provide an API for the interfaces, as well as a DLL.

Before You Start

Read all documentation. Make sure your plug-in idea follows all the constraints presented by the documentation. In addition, be sure to have the .NET Framework 2.0 installed, as well as a C# compiler. Visual C# Express is free and readily available through the Microsoft website.

Example

The available plug-in types for MyProgram are the ISensor and IDataFilter plug-in types. Making a plug-in that requires no hardware interfacing at all should be a primary test. As such, the RandomSensor idea was born. RandomSensor will return a random number upon a request for a value. The documentation would show that it requires 4 MrHomerPlugin Attributes: Name, Description, Make, and Model. A skeleton for the RandomSensor class would be something like:

```
[MrHomerPlugin("Name", "Software Sensor: Random")]
[MrHomerPlugin("Make", "Fake Sensor")]
[MrHomerPlugin("Model", "Wun-2-fif-T")]
[MrHomerPlugin("Description", "I am a description. Rawr!")]
public class RandomSensor : ISensor
{
   //Implementation details
}
```

Program It!

Programming the plug-in, once you have the shell is as simple as implementing any other interface. However, it is important to strictly adhere to all documentation provided about the plug-in type. If it expects something to be returned and used later, do not just return null!

Example

RandomSensor, as an implementation of ISensor, is required to implement the following:

```
double RequestData();
string GetUnits();
byte[] SerializeConfiguration();
void DeserializeConfiguration(byte[] data);
```

The first two methods are simple enough. The first checks the sensor's current readout and simply returns it. In this case, it returns a random double. The second gets the units of measurement. Once again, this is simple. Return a unit of measurement in string form. The third and fourth methods are a bit trickier. The documentation would show that if SerializeConfiguration were called, and the byte stream was passed across a network, if a RandomSensor there was called with DeserializeConfiguration(data), it would have the same configuration as the RandomSensor given. Essentially, the purpose of those methods is to enable a user to save a sensor's settings and restore them later. For RandomSensor, no

settings need to be saved, and as such, no real serialization needs to occur. Presented below is a sample RandomSensor class.

```csharp
[MrHomerPlugin("Name", "Software Sensor: Random")]
[MrHomerPlugin("Make", "Fake Sensor")]
[MrHomerPlugin("Model", "Wun-2-fif-T")]
[MrHomerPlugin("Description", "I am a description. Rawr!")]
public class RandomSensor : ISensor
{
  private static int number;
  private string name;
  public RandomSensor()
  {
    name = "Random" + (RandomSensor.number++);
  }

  public DataReading RequestData()
  {
    return new DataReading(name, (new Random()).NextDouble()*50);
  }

  public string GetUnits()
  {
    return "Marklar";
  }

  public void SetName(string s)
  {
    name = s;
  }

  public byte[] SerializeConfiguration()
  {
    return (new UTF8Encoding()).GetBytes(name);
  }

  public void DeserializeConfiguration(byte[] data)
  {
    name = (new UTF8Encoding()).GetString(data);
  }

  public string Author { get { return "Sukari"; } }
  public string Version { get { return "1.2"; } }
  public string Label { get { return "Random"; } }
}
```

Test It

Compile the plug-in into a DLL. Several plug-ins of varying types may be compiled into a single DLL. Once the plug-in is compiled, refer to the API to see how it should be loaded into the program. Usually, it is as simple as dropping it into a folder or loading it through an in-program command. However, be prepared to dig through configuration files as well. This step is entirely up to the programmer. Test your plug-in thoroughly to make sure it works as intended. It is

recommended that a small sample program be written to make sure every method returns what is requested. Sample output from an example program is below.

A small sample console application was written to test RandomSensor. The output below shows that RequestData worked as intended:

```
40.218167561208
3.30286265970341
22.9903494813435
```

At this point, it is advisable to test the plug-in within the program. RandomSensor in this case would be working in a test application to test sensor management. The sample output is shown below.

```
Obtained Sensor Manager.
Obtained Plugin Manager.
Obtained Database Manager.
Registered 'Fake' Database Type.
Connected to Fake database 'lol'.
Interfaces loaded from DLL.
Interfaces loaded from current assembly.
Plugins loaded from DLL.
Plugins loaded from current assembly.
Sensor and Filter Plug-ins loaded.
Available Sensor Types:
--Software Sensor: Random
----Fake Sensor
----Wun-2-fif-T
----Not to be used in any real examples. Only a test.
Available Filter Types:
--Fake Database Filter
----NOT 4 REELZ
Attempting to create a sensor...
Successfully added sensor!
Attempting to add filter...
Successfully added filter!
Attempting to start sensor...
Successfully started sensor!
10.1245973539188
23.2092924524142
44.4750019090599
15.7407113657057
37.0064208223514
Attempting to stop sensor...
Successfully stopped sensor!
```

As shown above, the test plug-in worked successfully. Real plug-ins may be written using the provided skeleton, with changes to both the attributes and to the class name.
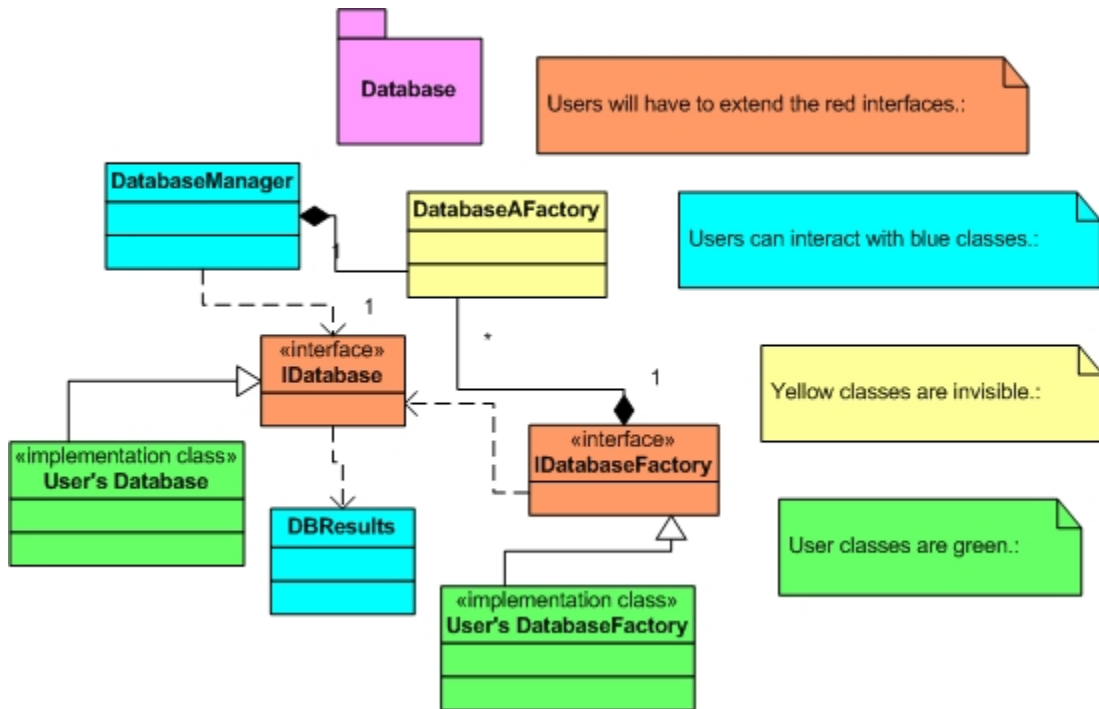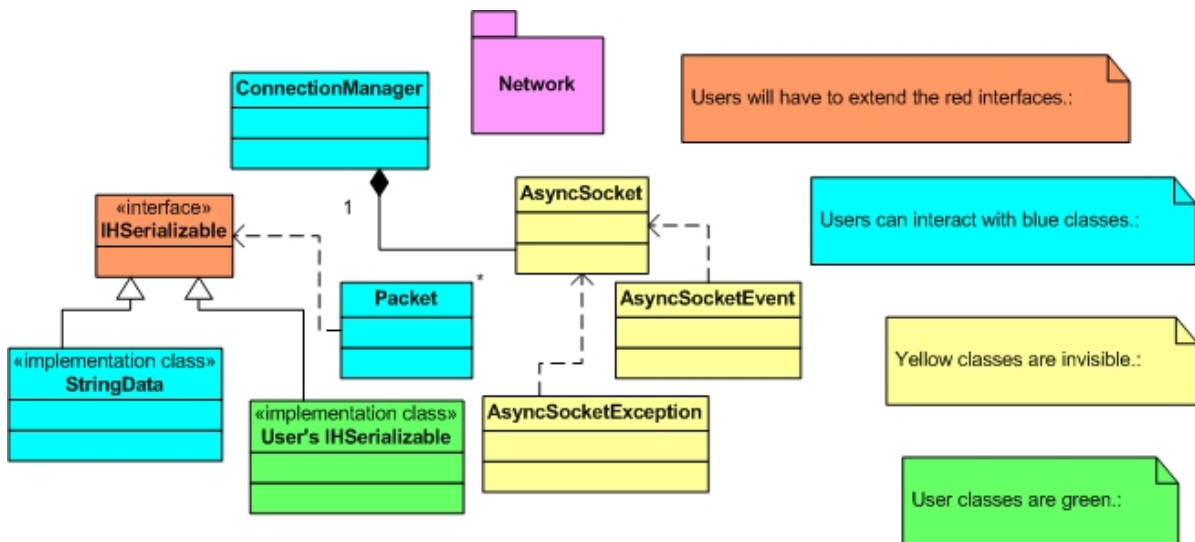
Appendix Q: UML Diagrams
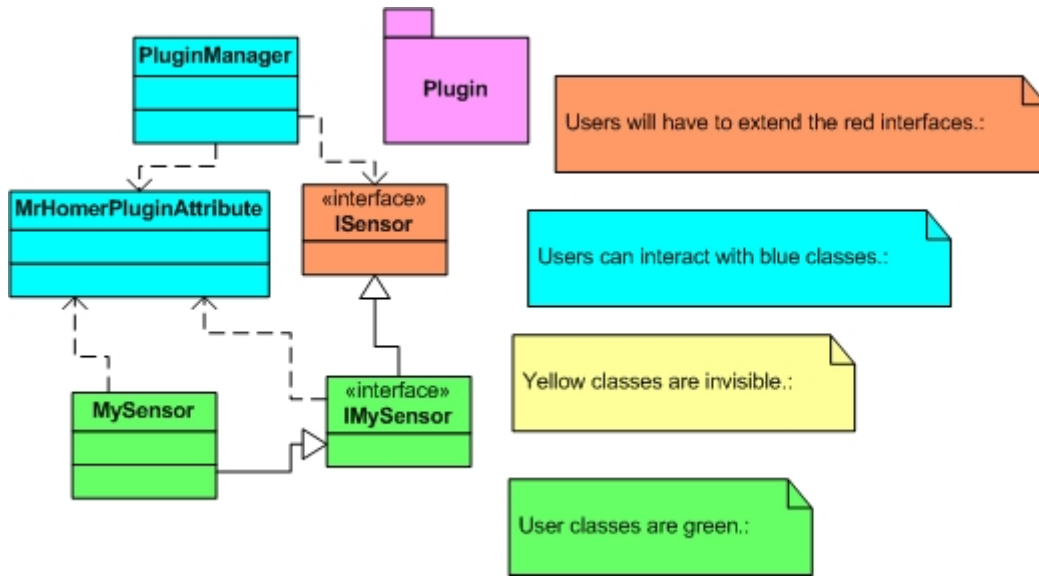
Figure N: Database
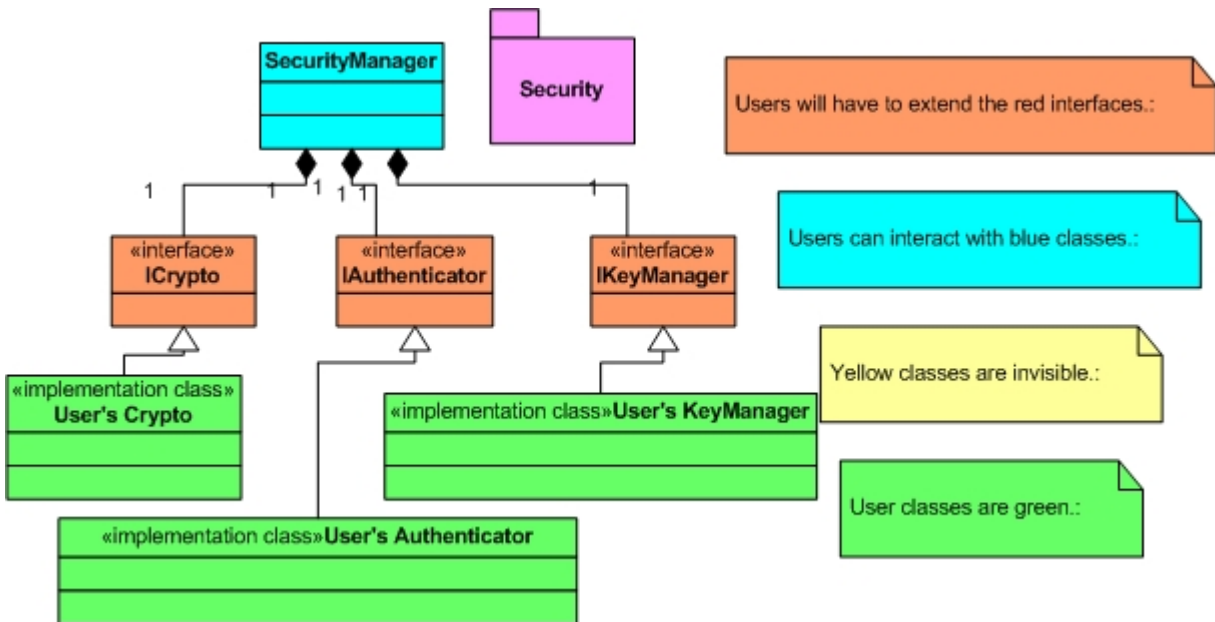
Figure O: Network

Figure P: Plug-Ins

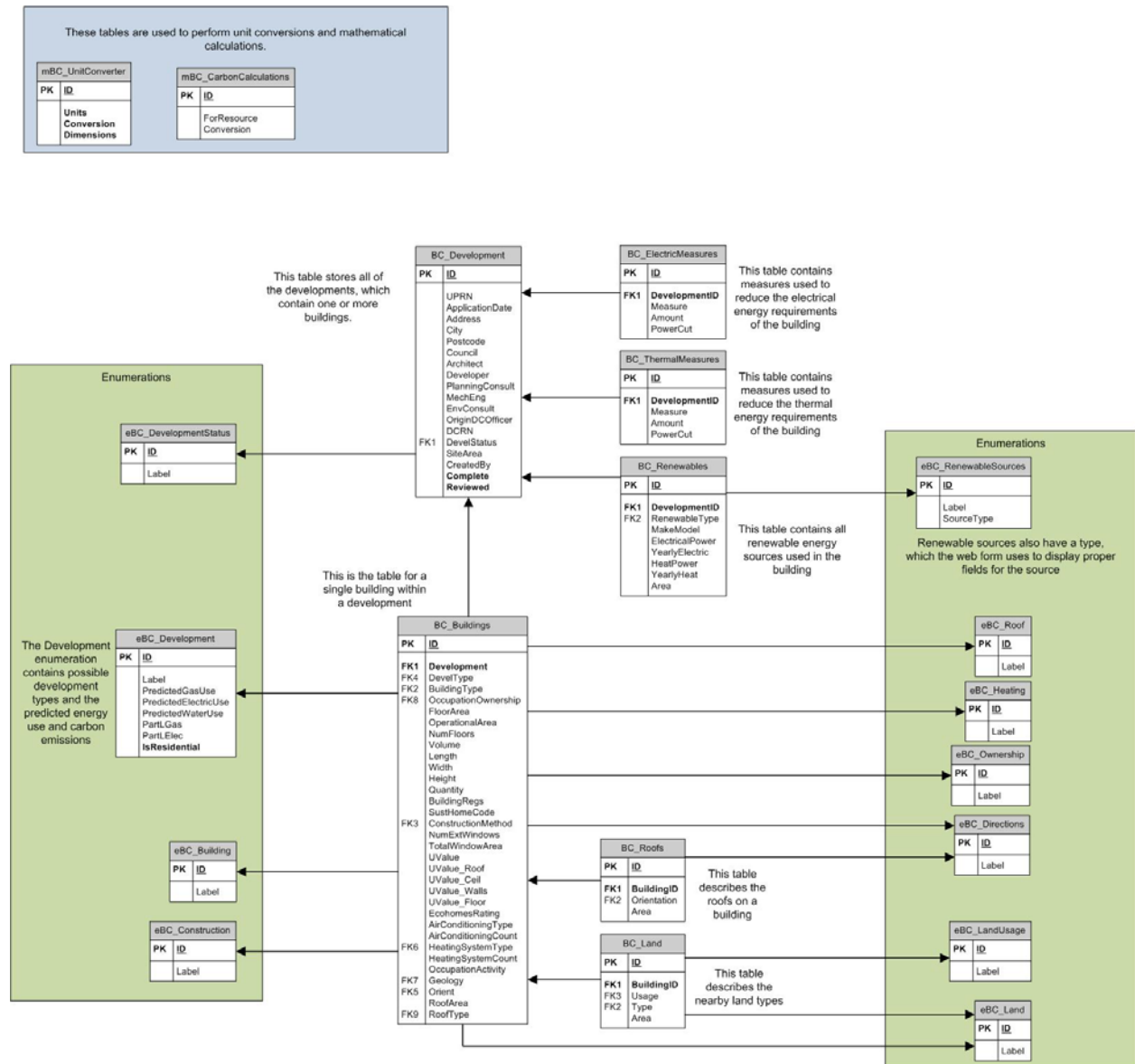**Figure Q: Security**

Appendix R: Database Diagrams
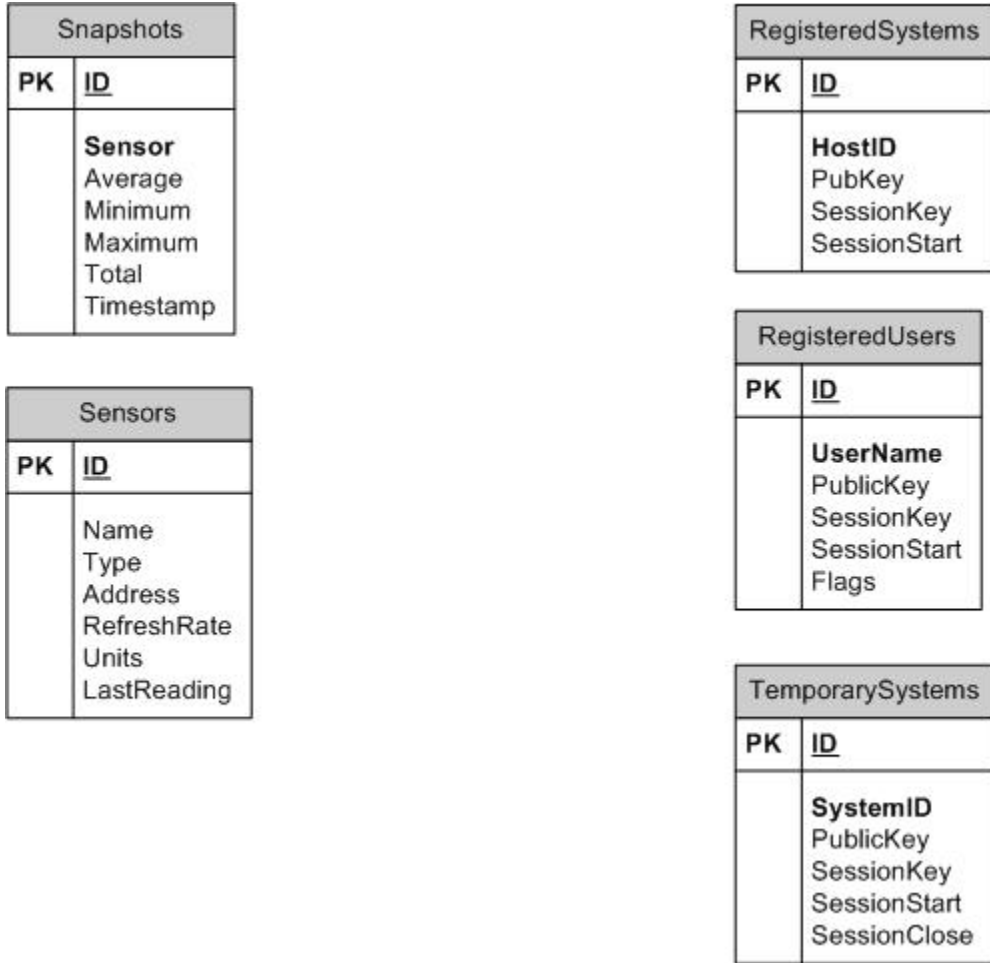

Figure R: RELIEF Database

Figure S: Energy Reporting Backend Database

Appendix S: Energy Reporting Backend Software Architecture Document

Introduction

MrHomer is a framework designed and implemented by the C-2008 Merton IQP Team. This document is intended for use by future developers (specifically, future IQP teams) for expansion and use. It will provide insight as to the system's design and proper use, as well as provide rationale for many of the design decisions made by the system's developers. MrHomer is a C# development. C# was selected as the primary language of choice because it allows for rapid development; Because MrHomer is being used to develop a prototype and the final product is on a very strict and inflexible timetable, rapid development was an important consideration. In addition to C#, Visual Basic .NET may also be easily used to use and expand upon MrHomer. However, no tutorials or guides will be provided for Visual Basic .NET.

Purpose

The purpose of MrHomer is to provide a solid base from which all future code expansion can be performed without needing to rewrite large portions of code. The result is a code base that enables users to implement almost any feature needed without needing to alter or recompile the framework.

Definitions and Acronyms

- ERB: Energy Reporting Backend
- IQP: Interactive Qualifying Project
- WPI: Worcester Polytechnic Institute
- RUP: Rational Unified Process
- Package: A group of software components designed to work together
- Class: A blueprint for a software structure consisting of behavior and data
- Object: The actual data and behavior constructed from a class
- UML: Unified Modeling Language

Architectural Representation

The architecture of ERB will be represented in the 4+1 Views' perspectives as recommended by the RUP. The 4+1 Views model includes a logical perspective, a development

perspective, a process perspective, a physical perspective, and a scenario perspective. Each perspective is described in detail below, and each perspective's application to ERB is provided a section later in the document.

Logical View

The logical view provides an overview of the packages that the system provides. In addition, brief details will be given about each package.

Development View

The development view provides a detailed overview of what a developer would working with. It expands upon each of the packages described in the logical view.

Scenario View

Because ERB is an application-specific piece of software and not an API or a framework, this section will be omitted.

Physical View

The physical view would provide an overview of how the ERB system integrates with a computer's existing hardware, as well as any considerations the system's users will need to take. ERB , at the time of writing, has no major hardware constraints, and as such the Physical view will not be included in this document.

Process View

The process view will describe in detail the various components that may run concurrently. In addition, it will provide insight into deployment of ERB, its efficiency, as well as its potential scalability.

Architectural Goals and Constraints

Goals:

- Automatic sensor updating. "Set it and forget it."

- Simple sensor importing.

- Automated communication system.

- Provide all information a GUI writer may want to display.

- Provide method by which outside programmers would update the system without seeing its source code.

- Provide methods for a user to tap into the automated communication system.

Constraints

- Must be thread-safe.

- Must make full use of MrHomer.

- Must wrap most of MrHomer for application-specific purposes, but still be compatible with it for more advanced functionality.

Logical View

ERB is divided into several packages, each of which is designed to provide simpler access to MrHomer functionality, or to apply program-specific functionality to pieces of MrHomer. The ERB packages are Main, Communication, Sensors, Messaging, and Database.

Main

The Main package provides simplified system initialization, as well as simplified system access. Several packages within ERB make use of the functionality provided from within Main. Main depends on all other packages at it acts as a central portion of ERB.

Communication

The Communication package encompasses the functionality of the ConnectionManager from MrHomer, as well as the SecurityManager. It is responsible for receiving packets and handling them appropriately. It depends on the Database package, as well as the Messaging package.

Sensors

The Sensors package handles sensors, be they hardware or software. Given the undefined method of receiving sensor data, this package makes heavy use of the PluginManager from MrHomer, allowing all sensors along with interpretation methods for sensors to be done entirely through plug-ins. Individual data interpretation modules may depend on the various other packages, and may even depend on MrHomer.

Messaging

The Messaging package handles all packet interpretation and automatically sends appropriate replies. The individual implementations of various Message Handlers depend on the Communication package, but due to their design as plug-ins, there is no true cross-package cyclic dependency. Several also depend on the Database package. This system utilizes the MrHomer PluginManager.

Database

The Database package wraps the MrHomer Database package with several program-specific data table entry classes, along with structures that contain database, table, and column names.

Development View

ERB 's packages are not all directly used. Several of them simply provide interfaces through which developers may add functionality and contain realizations of those interfaces.

Main

Brewery

Brewery provides a central data hub for all communications throughout ERB . It is also the class that will initialize all of MrHomer and ERB . As such, the first thing to do in any ERB program is to call the Brewery through its singleton implementation.

SearchCache

The SearchCache will contain a list of all searches that have been performed, thus allowing a user interface to depend less upon sending out request packets to the network chain.

ErrorManager

All errors that occur within ERB are put in the ErrorManager, allowing a UI writer to display them appropriately or to log them in a file.

Communication

CommunicationManager

The CommunicationManager is a task queuing service that allows a user to queue network tasks. It is thread safe.

ITask

ITask is essentially a task interface for the CommunicationManager. There are several tasks already implemented, including a task to initiate a network connection, a task to send data, and a task to disconnect. In addition, several tasks always run, including a data reading task, a listening task, and an authentication task.

Sensors

SensorManager

The SensorManager is the class through which sensor types will be accessed, and sensors will be created. In addition, filters are added through the sensor manager. In addition, the sensor manager is responsible for starting and stopping sensors.

ISensor

ISensor is the interface through which sensors are implemented. ERB includes one sensor as a test, the SoftwareSensor.

IDataFilter

IDataFilter is the interface through which data filtering is implemented. Each sensor has any number of filters attached to it. ERB includes two filters: the Database Filter, which puts data in a database, and the Database Cleanup Filter, which condenses database filter entries into snapshots.

SensorInformation

The SensorInformation class holds several pieces of data about the sensor in question, including the make and model. This information would be presented to the user of the system in the user interface, allowing someone to configure it properly.

FilterInformation

The FilterInformation class holds several pieces of data about the filter in question, including the make and model. This information would be presented to the user of the system in the user interface, allowing someone to add the appropriate filters to a sensor.

Messaging

IMessageHandler

Message Handlers handle messages that the Communication Manager received. Each handler has a subject associated with it, and packets with the given subject are then passed to the appropriate message handler. ERB comes with an extensive list of MessageHandlers already implemented. See the source code for details.

Database

TableEntry

There are several table entry classes, each for one of the ERB database's tables. Each one provides static methods to access column names, along with methods for finding entries in the table and inserting values.

DataTable

Should additional functionality with databases be needed, the DataTable class provides capability to browse an entire table, rather than a single entry.

DuffbeerDatabaseConfig

DuffbeerDatabaseConfig is a static class that provides access to the database and table names; allowing the compiler to catch errors when using them, as well as a single place to consistently store the information for use within ERB.

Process View

There are a few notes that need to be made about ERB, specifically with respect to Threading, Deployment, and Scalability.

Threading

ERB, on the surface, is procedural. That is, ERB can be used by anybody who knows how to write simple, one-process code. However, underneath the simple API lies a multi-threaded system with constant update mechanisms. ERB 's internals are entirely thread-safe.

The developers recommend several ERB functions, such as networking, be running on their own threads. However, it is not necessary.

Deployment

ERB has several requirements. Simply, ERB requires the .NET Framework 2.0 from Microsoft to be used. In addition, programmers using ERB need three things.

- A .NET language compiler, such as Video Studio .NET, or Visual C# Express
- MrHomer.dll
- Duffbeer.dll

Scalability

ERB 's individual components are highly scalable. Given an increased workload, they will function just as expected, with no major differences in processing time.

Appendix T: User Interface Guidelines

Abstract

This document provides a set of guidelines for the future implementation of the MrHomer user interface (henceforth called DONUT [acronym pending]) and a web-based viewer (DOH [Dynamic Online Homer). The naming is consistent with the theme of the other components. This document also covers RELIEF-web.

DONUT

The main task of DONUT is to display live information from the system running MrHomer. In accordance with the MrHomer design goals, DONUT must provide the following features:

- Provide a list of all connected sensors

- Display the most recent reading of each sensor

- Display a list of connected clients, if applicable

- Display a list of upstream neighbors, if applicable

- Display the ratio of renewable energy to non-renewable energy

In addition, the following features are optional, but recommended:

- Graph display of readings from each sensor versus time

- Ability to traverse the network tree

- Integration with the Merton LOUIS

- Link with RELIEF

These features serve only as content guidelines. The design of the UI features is not covered. Figure T shows a suggested presentation. Implementers should note that all interaction with MrHomer is through the Duff Beer package. The Brewery exposes all needed functionality; there is no need to access Duff Beer's utilities directly.
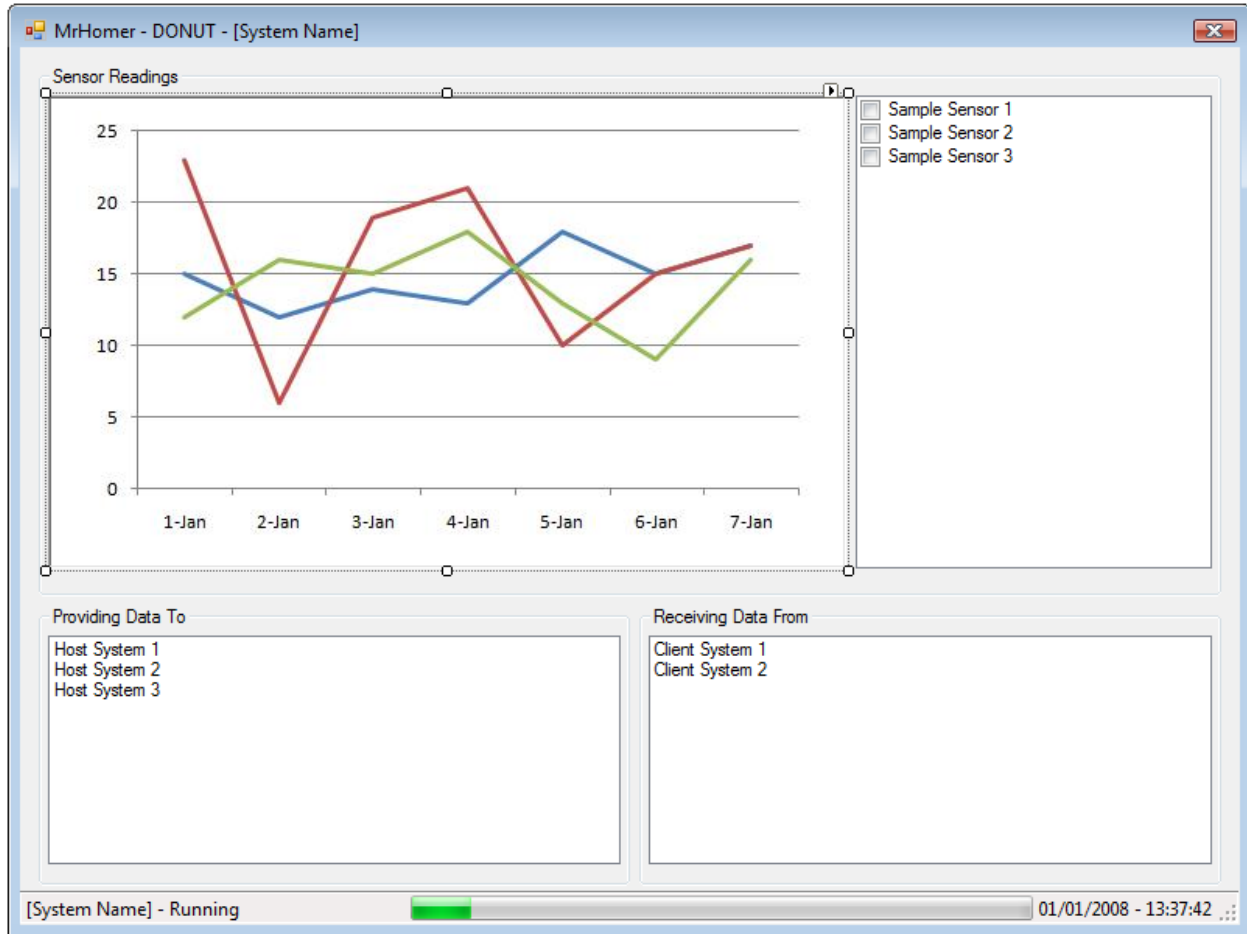
Figure T: Sample DONUT Window

DOH

A specific request was that the system be viewable from anywhere. This request prompted the design of DOH. When the user accesses DOH, it will prompt the user for credentials. The user will log in, and DOH will consult MrHomer for validation. Assuming a valid login, the user will have access to the following features.

- Provide a list of all connected sensors

- Display the most recent reading of each sensor

- Display a list of connected systems (upstream or downstream), if applicable

- Display the ratio of renewable energy to non-renewable energy

- Ability to traverse the network tree

- Integration with the Merton LOUIS and RELIEF

Note that Integration with LOUIS is mandatory for DOH. Although this is optional in DONUT, Mr. Adrian Hewitt explicitly requested its inclusion. In addition, DONUT may be integrated into DOH, creating a system entirely manageable via a web browser, removing the need for additional user interface software. Such an implementation can be achieved using only the resources provided in Energy Reporting Backend and MrHomer.

RELIEF

RELIEF contains the minimal set of required fields. Future users may add additional fields, but may remove none. This requires modification of the underlying database. The developers strongly advise that the current visual presentation (color-coding dropdown boxes red and text boxes green) not remain in the final version.

Appendix U: Sample Planning Application with Condition 38

PLANNING APPLICATIONS COMMITTEE   Item No: 2
13th December 2007

UPRN	APPLICATION NO.	DATE VALID
57/22/1623/027	(1) 07/P1347	24/04/07
	(2) 07/P1344	24/04/07

Address/Site   The Former Atkinson Morley Hospital and The Firs, Copse Hill, SW20

(Ward)   Village

Proposal: (1)   Alterations, extensions and new buildings in association with continued use of the site for hospital purposes (new private hospital). Alterations to existing vehicular and bus access and creation of a new bus turnaround. Provision of associated (131 spaces), landscaping, a cycle and footpath and sports fields. Retention of 27 car parking spaces off Cottenham Park Road for use by the Wolfson Rehabilitation Centre. Demolition of extensions and outbuildings associated with the hospital.

      (2)   Conservation area consent for demolition of extensions and outbuildings in association with the former hospital.

Drawing Nos   AR/AN/GA-01, /MB/GA/TPA-11 E, /TPA-18 B, /TPA19 B, /TPA-20 B, /TPA-21, /FD/GA-01 B, /GA-02 B, /GA-03 A, /GA-04, /SP/TPA-08 D & Site Location Plan

Contact Officer: Michaela Mercer (0208 545 3981)

RECOMMENDATION
(1) GRANT Planning Permission subject to the completion of a Section 106 Obligation and conditions
(2) GRANT Conservation Area Consent subject conditions

1.   **SITE AND SURROUNDINGS**

1.1   The application site is approximately 9.95 hectares in area. It fronts onto Copse Hill to the north and extends southwards to Cottenham Park Road and Camford Place, West Wimbledon.

1.2   The application site comprises the area occupied by the buildings of the former Atkinson Morley Hospital (Class C2) including the former nurses accommodation at the Firs. Whilst it includes part of the site containing the Wolfson Rehabilitation Centre it does not include any part of that building which is to remain in operational use.

documentation and evidence that the agreed site remediation criteria have been met.

Informative: The validation report is to be accompanied by a letter signed by a senior member of the development company and the consulting engineer or environmental consultant confirming that the site has been remediated in accordance with the agreed strategy

Reason for condition: In order to safeguard the health of future occupiers of the site and neighbouring areas.

36. The external sports facilities shall not be used outside of the hours of 08.00 – 22.00 without the prior written approval of the Local Planning Authority.

Reason for condition: To protect the amenities of occupiers of noise sensitive receptors and neighbouring residential properties.

37. No loading or unloading of goods or refuse or other servicing activities shall take place at the site outside the hours of 23.00 – 07.00

Reason for condition: To safeguard the amenities of the occupiers of noise sensitive receptors and neighbouring residential properties

38. No development shall take place until the Local Planning Authority has approved a report provided by the applicant identifying how the predicted CO_2 emissions of the development will be reduced by at least 10% through the use of on site renewable energy equipment. The developer shall install broadband/ wireless remote sensors so that Merton Council (or an approved organization) can monitor the energy usage of the building, and/ or the performance of the renewable energy equipment. Before any unit is occupied or sold the renewable energy requirement shall have been installed and the local planning authority shall be satisfied that their day to day operation will provide energy for the development for so long as the development remains in existence.

Reason for condition: To comply with LB of Merton UDP Policy PE.13, the Mayors London Plan policy 4A.7 and Energy Strategy proposal 13, the revised London Plan (draft), Planning Policy Statement 22 (Renewable Energy).

39. Amended Vehicle Access

Prior to the commencement of the development details of the proposed amendments to the vehicular accesses from Copse Hill to serve the development, including details of tactile paving and any associated amendments to parking control measures, signing and lining within the site boundary and on the public highway, shall be submitted to and approved in writing by the Local Planning Authority. Such works to be completed prior to the first occupation of the development.

Reason for condition: To ensure the provision of satisfactory access

Appendix V: Weekly Reports

Week 1 (01/07 - 01/13)

To make sure we started on the right foot we began by discussing our project with our liaison, Adrian Hewitt. In order for us to synchronize with him, he showed his Power-Point presentations to us and discussed his situation and expectations. We, in turn, gave our presentation to demonstrate our current understanding of the project. After collaborating, we developed a rough timeline to work out our major project goals.

Deeming it more efficient to split up according to our academic majors, the Alpha and Beta teams were formed. The Alpha team, consisting of Robert Bernard, Civil Engineering, and Benjamin Lessard, mechanical Engineering, were to concentrate upon the social and communications aspects. In the meantime the Beta team, Christopher Scalabrini, Computer Science, and Brian O'Keefe, Electrical and Computer Engineering, worked to develop the technical side of the project. Each team developed a timeline to outline specific tasks they needed to accomplish. The end goal of each of these outlines was to synchronize back into a single group, the Gamma team, and polish off the end projects together.

The Alpha team began research into renewable energy sources to gain an understanding of the pros and cons of each type. These include photovoltaic panels, wind turbines, combined heat and power generators, solar hot water heaters, and geothermal water heating and cooling. The Alpha team looked into current prototypical Building Birth Certificates as well and worked on revising them to reduce loopholes (e.g. units). Research was also conducted into the 1APP application system, a system where building permits are applied for and reviewed for approval on-line instead of using paper copies. Alpha team also spoke with the head of the Development Control Department and developed a set of questions to help further understand the system. Since this is to become the only way to apply for a building permit beginning April 6, 2008, the Team will also look into the feasibility of entering the Renewable Energy Information Form (Building Birth Certificate) as an application in this system.

The Beta team wrote a data-reporting program that could work via a network. It enabled one computer to share data it was receiving with another computer. This mock-up was a small-scale demonstration of the long-term goal of the Merton project center where databases would share information in a City Knowledge system.

To record progress made accounts were established with WPI's SourceForge. This website allows the uploading, reviewing, and editing of work done. This made it much simpler to work as a group as well as enabling our liaison to observe our progress with ease.

Week 2 (01/14 - 01/18)

Alpha Team

The Alpha team started the week revising the current Building Birth Certificates already in use by the Planning and Projects Department. To determine how to improve them a meeting was set up with Ed Cotterill. As a Naval Engineer working on his Masters thesis in a similar field, and as an asset to Adrian Hewitt in the writing of the original Building Birth Certificates, his opinions would be a significant help. Incorporating his suggestions with the Team's ideas a more complete Building Birth Certificate could be created. Mr. Cotterill also volunteered to inquire among his colleagues about the 1APP application system and get back to the Team with any further information.

Alpha team was also able to make contact with the Big Yellow Self Storage Company regarding a meeting to discuss participation in a test run of our energy reporting system. Upon brainstorming, the Team also decided to use the wind turbines on top of the Merton Civic Center as another possible test bed. A discussion with David Bell, the Energy and Sustainability Manager, was set up.

Contact was made with the upcoming WPI Merton Team Two. Team One offered to set up a conference call to discuss any problems with them. To further assist continuity with future groups, Team One also put together a complete set of preliminary guidelines for Team Two and Team Three.

Beta Team

Contact was made with Dr. Fabio Carrera, the originator of the City Knowledge concept. Beta team explained the plan for the proposed system, including database design and the idea for each building to store its own database. Dr. Carrera pointed out some minor flaws in our design, such as things missed in the database design. In light of Dr. Carrera's suggestions, minor revisions were made to the database design plan.

Work is progressing on the program framework. Currently, the frameworks for network communications, security, and database access have been completed. What remains is to

translate the database plan into an actual database and to complete the communication interface between two systems.

Week 3 (01/21 - 01/24 and 01/29)

Work continued on the building birth certificates to improve usability and relevancy. Different entries were separated by type for improved organization. The Team began constructing the preliminary RELIEF (Renewable Energy Localized Installation Entry Form) user interface. To keep the interface simple and effective pages were constructed for relevant sections. The general project information would be inserted in the opening page and a link would be selected for each additional development in the project. This will simplify the process especially for contractors with mixed developments.

A more relevant preamble was put together to use at interviews and meetings. The preamble briefly describes the Team the goals of the project. This will provide an introduction for anyone unfamiliar with the project. With this, a quicker and more productive meeting/interview will occur due to a common understanding.

Contact was made with David Bell, the Energy and Sustainability Manager for the Civic Center. The Team spoke with him regarding the possibility of using the wind turbines atop the Merton Civic Centre as a test bed for the project. He agreed that this could be done and mentioned that his department could finance the necessary energy-monitoring device up to a few hundred pounds. He also mentioned a wind turbine station in the Borough of Southwark that would be worth visiting. After speaking with him, he took the Team up to the 15th floor to see the actual turbines and the possible places to put the equipment that would monitor them. He also provided a few contacts to find a monitoring device.

A meeting with Russell Lerman from ENVIDO was also set up. He is available on Friday to obtain a monitoring device that will work with the Wind Turbines and the new program.

By David Bell's suggestion, the Team sent a preliminary email to Declan Stegner to see if he had an available old PC to manage the monitoring device. Upon meeting, the Team will also ask a few different things including GIS availability and the possibility of connecting the monitoring computer to the Merton network.

Contact with the Borough of Southwark was made in order to set up a meeting to visit the wind turbine station.

The Team set up a preliminary test run of the RELIEF. A member of the Planning and Projects Department in Merton agreed to go through the form as if they were a contractor filling it out. Although the form was still incomplete, a few bugs were located and eliminated.

The framework for the reporting program was finished, tested thoroughly, and is now operational. The next step is to write the back end (do what the program is supposed to do, reporting, connecting, updating).

RELIEF is entering the final stages of completion. The form is fully functional but still needs some small updates and changes. Testing and usability study should occur by next week, also creating a list of idea/guidelines for E Term for the user interface.

Week 4 (01/30 - 02/01)

The Team met with Kelly Howell, the Planning Systems Manager, on Friday and discussed the possibility of having RELIEF in the 1APP system. She seemed skeptical of getting it into the system because of the national standards. The questions asked on the local level can only come from a predetermined list; the questions needed for the 10% rule are not on this list. She also said that it would be difficult for Development Control to get the developers to fill out RELIEF because within 1APP, there is nothing that legally would require a developer to complete the form. After the meeting, possible solutions for this problem were discussed. In addition, in the meeting Kelly explained some of the forms, and most of the process for 1APP.

A major accomplishment for week 4 was the completion of RELIEF. Currently, there is a live version of RELIEF, and the Team is in the process of constructing a usability study. Now that a preliminary version of RELIEF is available, the Team will introduce further refinements and implement some of the requested features. The usability study so far has identified three small problems with the form. To eliminate the confusion experienced while copying the system to the test server, the Team will also plan a RELIEF setup package.

Using the previously developed MrHomer framework, a large portion of the next portion of the program was produced, dubbed the Energy Reporting Backend. In addition, we produced a User Guide for the Plug-In system of MrHomer, a simple piece of software. In addition, the team focused on documenting all code thoroughly, as future teams will likely need to look at it. At the time of writing, all code in both program layers is fully documented and functional.

Plans for this week include working on the communications portion of the Energy Reporting Backend. The communication portion will entail actual communications protocols, security, and the network's ability to grow. In addition, the Team plans to speak with several software engineers to determine the clarity of the user guide and the SAD.

The Team met with Declan Stegner, the head of the Information Technology team and Merton GIS manager. He was able to secure an old computer for the Civic Center test bed. Contact was also made with his neighbor and colleague Gary Shaw, the GIS Consultant. He explained how UPRNs are assigned. The building first applies for construction permission through the Development Control Department and Planning and Projects Department. Only after the construction is finished will Declan Stegner be notified of its location. He then assigns it a Unique Property Reference Number. These numbers are simply sequential and are assigned in the order completed (i.e. 48000101, 48000102, 48000103, etc.). The only set numbers in the Merton UPRNs are the "48" at the beginning to denote which council the building is located in. Each council is assigned a different classification number to improve organization inside the national GIS to which each council's GIS must report to.

The Team also met with Russell Lerman, of the energy monitoring company Envido, to discuss power meter possibilities for our test bed using the Merton Civic Centre wind turbines. David Bell again granted access to the 15th floor to see the current meter readouts. Upon inspection, Russell concluded that these were the wrong type to connect directly with a computer. He suggested the use of a few wireless monitors of his that could connect with a data logger he was installing on the sixth floor that would connect directly into the Internet. After mentioning the preference of using a hardwire directly into a computer to act as a server near the sensors, Russell thought this would not be as effective a method. Dave suggested implementing a power meter with two outputs to try both methods and compare our results. Russell also demonstrated the website end of his data logger system which is very similar to the Team's long-term project. It provides secured levels of access and organized by level (i.e. sensor, building, borough, city). He will reply soon with monitor prices and capabilities within the next few days. After selecting one, the Teams will make contact with him again to schedule an instillation.

Week 5 (02/04 – 02/08)

On Friday, the Team met with the Big Yellow Self Storage Company. Adrian and the Team went down around 2:00 PM to investigate the PV's on the roof and the set up they have for gathering the energy. We discussed the methods we have been looking into for monitoring and explained the project to them. They were very excited to see where the project will go. They were willing to set up a test bed when Team Two gets to Merton.

Week 5 saw the preliminary completion of RELIEF-web. This version incorporates suggestions from respondents of the needs study. The Team sent an email to the needs study user list requesting review of the prototype forms. The Energy Reporting Backend is also nearing completion. It now includes most of the database features and sensor support. Remaining are the advanced security features and the communication features.

This week work continued on the final report. All of the necessary sections were added and the various details available have been added at this point. Various details that were not addressed in the end of the week submission to the advisors will be finished next week.

Appendix W: Overall System Guidelines

This document outlines the operation of the complete energy reporting system. The intended audience of this document is the system developers. It details the Team's findings during the prototype development phase as they relate to full integration.

General

The London government desires a more intelligent and efficient way to gather energy reporting information. In addition, the Borough of Merton has suggested using the City Knowledge principles as a method of farming the data, versus the hunter-gatherer approach. As such, an automated system has been devised.

Reporting Program Feature List

- Reading data from any type of sensor (hardware, software)
- Interpret any type of input
- Distinctly tiered network structure
- User log-in system consistent across the network
- Database connectivity and use
- Updating utility to keep data up to date
- Flexible and extensible communication protocol
- Secure data transfer to protect against packet sniffing
- GIS Integration
- RELIEF Integration
- UPRN Linking
- Web-based User Interface and control panel

Overall Network Structure

The system would be a structured network of individual nodes, each node being a single data collection unit (with multiple sensors per unit). Each unit would be assigned to a building development. In addition, each Borough would have its own data collection unit, as would any larger governing body. The network would be structured as a directed tree. The building's data collection units would be the leaves of the tree, which would connect to the appropriate borough system. The borough systems would then connect to the higher system. Structuring the

program as a tree prevents bottlenecking. The tree could be further subdivided to help reduce bottlenecking.

Node Structure

Each node would be responsible for its own data collection, whether it collects data from sensors or from building below it in the tree. Each node would run identical software, with key differences being in that node's particular configuration. Running identical software will prevent the need to produce and debug multiple programs.

Users

Each node would maintain a set of permitted users. This would allow for privacy of information among individual nodes. In addition, users will either be flagged as cascading users or static users. A cascading user would be able to view data one level below the current level. The system would be responsible for adding cascading users to lower machine user caches. In addition, a machine further down may flag the user differently, such that a guest user would be able to see data at the national and borough level, but specific building details would remain hidden. A static user would only be able to view information on the current node. The benefit of cascading users is that it allows government officials at the national, regional, borough, or city level to view the statistics of buildings, as well as the owners of said buildings.

Nodes

Each node would also be able to locate the node above it, whether it is on the same network or over the Internet. In addition, each node would have a cache of permitted nodes below it. Each node is responsible for connecting to the node above it. As such, all non-tree nodes would require either a fixed IP address or a dynamic DNS. Any errors in connecting to the node above it would require contacting the IT professional maintaining the next node up.

Programming Language

The choice of programming language is important. While the system could be produced in many languages, the system contains some limitations that require consideration.

Object-Oriented Languages

The use of an object-oriented language would allow for flexible software architecture. In addition, the proper use of object-oriented concepts would allow extensions to be made to the system without major code changes. Given that the system will require regular updated with the

production of new sensors and with ever-changing data format requirements, proper use of inheritance and polymorphism will facilitate ease of development and ease of use.

Reflection

In addition to using object-oriented languages, a language that supports reflection is recommended. Since the program cannot possibly predict all sensor types and sensor code, the ability to instantiate data types and objects dynamically regardless of their type will allow for reusable code that will rarely need to be changed. In addition, a reflection package would allow Metadata to be used, which would allow a sensor programmer to inject sensor information directly in the code, which would remove the need for global variables, static classes, and unnecessary object instantiation.

Recommendation

The recommended language is C++. While the prototype was written in C#, C++ is not only more efficient, but is cross-platform. However, any of the .NET languages can be used, provided the system runs on a Windows environment.

Execution Environment

Execution environment is an important consideration. Each environment has its own advantages and disadvantages, highlighted below.

Windows

The prototype system was developed to run solely on a Windows-based system. Windows provides the benefits of familiarity to users running the system, as well as the .NET Framework, which could greatly increase development speed. However, Windows is not free, is prone to security issues, and consequently may not be the optimal choice.

Unix-like Systems

Unix-like systems, including Linux and BSD, have the advantage of being entirely configurable to whatever system specifications are required by the program. In addition, the system developer could choose to use the program as the system shell, thus removing the possibility of the user altering the system. In addition, these are free and are less prone to security issues. However, Unix-like machines are more difficult to properly configure.

Configuration File

Each node would have a configuration file that will identify the node and instruct it how to run. The program will refer to the configuration file in order to know which node is above it, as well as in order to know how to obtain all nodes below it. All things that could possibly vary between nodes will be specified in the configuration file, with emphasis being placed on designing an intelligent system versus a bloated configuration file. In addition, all hard-coded values, such as file paths, will be in the configuration file. It should be the only hard-coded value in the entire system, along with database table names and database table structures.

Database

The database is central to the reporting system, as it stores the users, systems, and sensor data. The reporting system should have the capability to support multiple types of databases.

Recommended Server

The database server should be transparent to the reporting system. It must also be selected for performance and reliability.

MySQL

MySQL is a free (under the Community AB Licenses) database server. It has good performance under most loads and is easy to set up.

MSSQL

Microsoft SQL Server is not free, but has the advantage of being supported by a large corporation. It runs only on Windows systems, but can be accessed directly from Visual Studio.

Database Structure

Any system using a database needs an intelligent database design. This section explains the recommended table structure for the database.

Security

There are four relevant tables for security: Users (for storing the registered users), Registered Systems (for systems that are always allowed to connect), Temporary Systems (for systems that can connect because they have been requested), and Key Storage (for security keys). The key storage table may be removed from the database if it is to be stored elsewhere (e.g. Kerberos authentication).

Farmed Data

Data from the sensors should be captured to a Snapshots table in the database. This will use both Sensors and Filters tables to store the configuration for sensors and filters, respectively, within the reporting program.

Communication

The only important table for communication is Connection Initiation. This table stores the addresses of everything the current system is to connect to upon startup.

Sensors

The vital portion of the system is the sensor architecture. As such, the system must be well designed and flexible in order to facilitate new sensor drivers being introduced in the system. Sensors are recommended to have a refresh rate and an identifier.

Plug-in System

It is recommended that the programmer implement a plug-in system in order to implement sensors. A plug-in system would require reflection to be part of the language, but would also allow for rapid development of sensor drivers suited to the system, as well as

minimizing changes that need to be made to the main program source code. In addition, by implementing sensors as plug-ins, sensors could be used in creative ways to implement other features, such as obtaining data from lower in the tree automatically.

Filter System

The programmer should use the plug-in system to devise a data filtering system. Data filters could be applied to sensors as a way of interpreting the data. For example, a filter could be designed that will take a composite of a hundred readings and use that as a snapshot, or to produce a graph. It is recommended that the programmer implements a Composite data filter that has a component for interpreting different types of data from sensors into a readable format, and a component for altering said data.

Serialization

The system, at any given time, should be able to save the software state of all sensors and filters for storage via serialization. By serializing the data and storing it, the program will be able to recall a previous sensor and filter setup and apply it at the program start-up. In addition, serialization of sensor information will allow the data to be transferred via network if another node requires knowledge of sensor status.

Data Storage

It is recommended that the data from sensors be stored in a database. A filter should be used to store snapshots in a database. Furthermore, it is recommended that each snapshot contain a maximum value, minimum value, a time span, and an average, for statistical reasons.

Auto-Update Mechanism

An automatic patching utility is essential to keeping files up to date. The system administrator at the top of the tree would broadcast that there is an update to all lower systems, which immediately connect to it and download it.

Patch Downloading

The patches would go into a separate folder, which would also contain information about the current patching history. Using said data, a system lower on the tree could request missed updates.

Peer-to-Peer

The advantage to peer-to-peer download systems is that the load is distributed among several network connections. The download connection is more robust and reliable. The only major disadvantage is that peer-to-peer networks have a negative image in the media due to misuse of the technology for digital piracy.

Direct Download

Direct downloads have the advantage of a known download host. However, direct downloads are less stable and less reliable. In addition, if the connection is fractured, the download is halted until the one server supplying the file is reconnected, and the file transfer must be reinitiated.

Recommendation

The project team recommends that a peer-to-peer system is implemented to distribute software updates as it is a more robust system and less prone to problems.

Patching Program Separation

The actual program that applies the patch is to be a separate program from the main sensor readout program. As such, changes cannot be easily made to the patching utility, since there is nothing to patch it. The utility will need to be robust, efficient, and flexible the first time around, and as such, thorough testing is imperative.

Recommended Program Behavior

The program, as far as patching is concerned, will determine when it has completed downloading a patch. At that point, the program will run the utility and terminate itself. Once the reporting program is terminated, the patching utility will patch the program, and will run the reporting system while terminating itself. If the patch has failed, the program will request the patch again and will continue until it receives an appropriate patch. A patch verification algorithm is required.

Security and Communication

As a distributed database network, computer-to-computer communication is vital; without it, the system simply would not exist. As such, the developer must take extra care to make sure that the networking component is efficient. In addition, since the data being transmitted is confidential, substantial security must be implemented.

Communication Protocol

The communication protocol should not be hard-coded. Hard-coding the protocol forced recompiles, and in addition, forces more frequent and longer updates. In addition, the user must not be capable of easily altering the protocol.

Plug-in System

The designing team recommends using a plug-in architecture combined with metadata in order to determine packet types. This would allow packets to be added as needed without excessive updates. In addition, all packets would use other packet's metadata in order to label the packets appropriately. This would prevent any small errors, and adhere to the rule against hard-coded values.

Packet Types

Packet types would come in several varieties.

- Data Request: A request to the client for data. All data requests should only cascade down the chain, never up.
- Data Response: The response to a data request. The lower server would encapsulate the data as a packet or a series of packets and send them to the server that requested it.
- Functional Packet: A higher system could send a packet whose sole purpose is to be correctly processed. Examples include forwarded packets that are sent to machines not directly connected to the sending machine, and informing servers that an update is ready.
- Credential Packet: Upon a successful network connection, a system lower on the chain will send a credential packet as a verification of its identity. The higher system would use the credential packet to determine whether or not to allow the lower system to access it.

Security

As all data is sent over an (probably) unsecured network, some form of security is required. A security protocol is needed to rectify this.

Key Management

Keys serve many purposes in security systems. These include identity verification and encryption. A key manager is needed to provide access to these keys. It should also be able to changes keys. A key cache is recommended to speed access.

Cryptography

The cryptography component performs all encryption and decryption. These are the only actions performed. Keys are provided from the key management components.

Authentication

The final security component is authentication. This component handles user and system registration, deregistration, verification, and access controls.

GIS Integration

The system is fully able to integrate into an existing GIS. The programmer is recommended to include building UPRNs into the database as system names. The UPRNs would allow a server to poll an existing GIS server for its data to provide it on demand.

RELIEF Data

The reporting system should be able to integrate with RELIEF, allowing users to view the baseline energy requirements for a given building. The RELIEF database should be connected in the Database configuration for access in the reporting program.

Real-Time Reporting

The system as stated is capable of real time remote data reporting. Due to the plug-in nature of sensors, a software sensor could be devised whose sole purpose is to remotely pull data off another server's database and put it on the current server's own database. Through such a system, pseudo-real-time reporting is possible.

Map-Based User Interface

There are two major parts of the user interface. The first one is the map-based user interface. Ideally, it would be a web-based user interface that would integrate with a web GIS system. Any information that is available when directly connected to a server should also be available via request from any lower system, giving a nearly seamless interface.

Data-Farmer User Interface

The other portion of the user interface is the sensor registration and filter application interface. The data-farmer user interface would be responsible for all the in-house maintenance, such as adding and altering sensors. Ideally, the data-farmer interface would be included in the web form, and access to it would be restricted on a per-user basis.

Input Devices

Input devices may play a large role in program design depending on whether or not the system is running as a server to which an HTTP client can connect. If the system is running on a web form, user input is irrelevant.

Mouse

The mouse should be the primary form of input, no matter what. It allows only simple types of input, which makes it much more difficult for a user to do something on accident. As such, most of the user's options should be selectable in drop-down boxes or via buttons.

Touch-Screen

The input is very similar to the mouse. Touch-screens are more intuitive and would offer that benefit to the computer-illiterate or those intimidated by computer use.

Keyboard

The user's only available keyboard input should be for search functionality, and only if it is necessary. Ideally, a developer would provide an intelligent searching system with pre-defined terms that the user would see, as the variety of search items is not large.

Error Handling

Errors must be handled as quietly and as non-intrusively as possible. Errors should be kept in a log that a system further up the chain could request, and program functionality must continue as completely as possible despite errors. In addition, any errors that are vital to the user (such as an abrupt disconnection of another system) should be displayed in a useful manner, such as highlighting the item in red.

Works Cited

British Telecommunications. (2008, January 01). BT. Retrieved January 22, 2008, from BT.com: http://www.bt.com

Carrera, F. (2004). *City Knowledge: An Emergent Information Infrastructure for Sustainable Urban Maintenance, Management and Planning.* Massachusetts Institute of Technology, Urban Studies and Planning. Boston: Massachusetts Institute of Technology.

Carrera, F. (2006, September 2). *City Lab @ WPI.* Retrieved November 8, 2007, from Fabio Carrera: http://ece.wpi.edu/CityLab/City_Lab.html

DeMello, J., Guzman, J., Luk, C. (., & Robert, M. (2006). *Carbon Reduction Tools for Municipal Buildings.* Worcester Polytechnic Institute. Worcester: n/a.

Department for Environmental Food and Rural Affairs. (2007, January 31). *e-Digest Environment Statistics, Climate Change.* Retrieved November 14, 2007, from UK Defra: http://www.defra.gov.uk/environment/statistics/globatmos/gagccukem.htm

Energy Information Administration. (2007, November 29). *Energy Emissions Data & Environmental Analysis.* Retrieved December 2, 2007, from Energy Information Administration: http://www.eia.doe.gov/environment.html

Energy Information Administration. (2007, May). *International Energy Outlook 2007.* Retrieved December 2, 2007, from Energy Information Administration: http://www.eia.doe.gov/oiaf/ieo/emissions.html

Environmental Protection Agency. (2007, November 30). *Combined Heat and Power Partnership.* Retrieved November 31, 2007, from U.S. Environmental Protection Agency: http://www.epa.gov/chp/

European Environment Agency. (2005, October). *Global and European temperature.* Retrieved December 1, 2007, from European Environment Agency: http://themes.eea.europa.eu/IMS/ISpecs/ISpecification20041006175027/IAssessment1116322231704/view_content

European Environment Agency. (2006, December 6). *Global Annual Average Temperature Deviations.* Retrieved December 1, 2007, from European Environment Agency: http://dataservice.eea.europa.eu/atlas/viewdata/viewpub.asp?id=2510

Fullerton College. (2007). *Enviromental Biology.* Retrieved November 13, 2007, from An Inconvient Truth: http://staffwww.fullcoll.edu/tmorris/an_inconvenient_truth/an_inconvenient_truth.htm

Le Treut, H., Somerville, R., Cubasch, U., Ding, Y., Mauritzen, C., Mokssit, A., et al. (2007). Historical Overview of Climate Change Science. In S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. Averyt, et al. (Eds.), *Climate Change 2007: The Physical Science Basis.* Cambridge, United Kingdom: Cambridge University Press.

London Assembly. (2005, October 26). *Pilot Energy Action Areas.* Retrieved November 14, 2007, from Mayor of London, the London Assembly, and the Greater London Authority: http://www.london.gov.uk/mayor/environment/energy/partnership-steering-group/energy-action.jsp

London Councils. (2006, November 29). *LH magazine: Domestic carbon emissions 2005.* Retrieved November 14, 2007, from London Councils: http://www.londoncouncils.gov.uk/doc.asp?doc=18839&cat=2443

London Energy Partnership. (2005, October 24). Energy Action Areas. *Working Towards Low Carbon Developments* . London, England, United Kingdom.

Marland, G., Boden, T. A., & Andres, R. J. (2007). Global, Regional, and National CO2 Emissions. *Trends: A Compendium of Data on Global Change* . Oak Ridge, Tennessee, U.S.A.: Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, U.S. Department of Energy.

Maunsell, F. (2004). *London Renewables Toolkit.* London: Greater London Authority.

Michaels, P. J., Balling, J. R., & Davis, R. E. (2005, October 10). *Sea Level Rise: How High?* Retrieved December 1, 2007, from World Climate Report: http://www.worldclimatereport.com/index.php/2005/10/10/sea-level-rise-how-high/

National Climatic Data Center. (2007, March 29). *Global Warming*. Retrieved December 1, 2007, from NOAA Satelite and Information Service: http://www.ncdc.noaa.gov/oa/climate/globalwarming.html#Q1

Natural Resources Defense Council. (2007, September 21). *The Consequences of Global Warming*. Retrieved December 9, 2007, from Natural Resources Defense Council: http://www.nrdc.org/globalWarming/fcons.asp

O'Hara, C., Hobson-Dupont, M., Hurgin, M., & Thierry, V. (2007). Monitoring Electricity Consumption on the WPI Campus. IQP Report, Worcester Polytechnic Institute, Worcester.

Privacy International. (1197, July 22). CCTV Frequently Asked Questions. Retrieved January 22, 2008, from Privacy International: http://www.privacyinternational.org/article.shtml?cmd[347]=x-347-61925&als[theme]=Video Surveillance&headline=CCTV Frequently Asked Questions

The Council of Merton. (2007, September 10). *List of Boroughs*. Retrieved December 2, 2007, from The Merton Rule: http://www.themertonrule.org/list-of-boroughs

The Council of Merton. (2007, October 11). *The Merton Rule*. Retrieved November 28, 2007, from The Merton Rule: http://www.themertonrule.org/

UN Department of Economic Affairs. (2005, August 11). *UN Department of Economic Affairs*. Retrieved November 10, 2007, from Agenda 21: http://www.un.org/esa/sustdev/documents/agenda21/english/agenda21toc.htm

United Nations Framework Convention on Climate Change. (2005, February 15). *Kyoto Protocol*. Retrieved November 26, 2007, from United Nations Framework Convention on Climate Change: http://unfccc.int/kyoto_protocol/items/2830.php

United Nations Framework Convention on Climate Change. (2007, October 11). *Kyoto Protocol*. Retrieved November 13, 2007, from United Nations Framework Convention on Climate Change: http://unfccc.int/kyoto_protocol/items/2830.php

United Nations Framework Convention on Climate Change. (2007, October 24). *National Greenhouse Gas Inventory Data for the Period 1990–2005*. Retrieved November 30, 2007, from United Nations Framework Convention on Climate Change: http://unfccc.int/resource/docs/2007/sbi/eng/30.pdf

United Nations. (2005, August 11). *UN Department of Economic Affairs*. Retrieved November 10, 2007, from Agenda 21: http://www.un.org/esa/sustdev/documents/agenda21/english/agenda21toc.htm

United Nations. (1992, May 9). United Nations Framework Convention on Climate Change. New York, New York, USA.

Weart, S. (2007, August 12). *The Discovery of Global Warming*. Retrieved December 9, 2007, from American Institute of Physics: http://www.aip.org/history/climate/index.html

World Resources Institute. (2003). Carbon Emissions from energy use and cement manufacturing, 1850 to 2000. World Resources Institute.