EVENT RECOGNITION FROM AMBIENT ASSISTIVE LIVING TECHNOLOGIES

by

ERIC WILLIAM BECKER

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

AUGUST 2010

ACKNOWLEDGEMENTS

I would like to acknowledge the support and assistance from the members of my committee. I want to thank Fillia Makedon, Roger Walker, David Kung, and Heng Huang for their patience and resolve while I worked on these projects.

I would like also to acknowledge Zhengyi Le and the other members of the Heracleia Human Centered Laboratory where this research was carried out.

I would also like to thank the co-authors of some of my publications, including Gutemberg Guerra-Filho and Vasillis Athitsos.

May 14, 2010

ABSTRACT

EVENT RECOGNITION FROM AMBIENT

ASSISTIVE LIVING TECHNOLOGIES

Eric William Becker, PhD

The University of Texas at Arlington, 2010

Supervising Professor: Fillia Makedon

As the population ages and technology advances, a need exists for creating ambient intelligent systems to be placed within the home environment. Attitudes towards technology have been changing, and home monitoring is now considered a less expensive and desirable alternative. Ideally, such systems should be small, wireless, and take the minimum of effort and cost to install and place within the home. In order to detect human activity in an assistive environment, key questions about the construction and operation of the technology and methods needed to detect that activity. To that end, a computational framework has been created inside an apartment testbed combining a variety of algorithms, tools, and methods that support an assistive living apartment using Wireless Sensor Networks and other devices and sensors.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter discusses the need of a computational framework of tools and methods for human activity monitoring and event recognition. The motivations for creating the tools and the experiments are given below.

1.1 Background

The germ of the idea behind this work began with the development of an Assistive Living Environment fitted with a Wireless Sensor Network. Combining the two components together would result in a test bed for running various different kinds of experiments to fit the needs of various researchers to produce results for study while maintaining a non-invasive environment.

1.1.1 Assistive Living

An assistive environment is a living or working space fitted with technology to aid a human in completing some job or task. In the case of assistive living, the technology is deployed to increase the quality of life of a human living within the environment. Sensors can sample the information, and automatic and robotic systems can then act upon the information to aid the human subject.

1.1.2 Wireless Sensor Networks

Today's assistive environments place pervasive technologies in the patient's living area in order to provide therapeutic or assistive type of human monitoring [1]. One type of pervasive technology commonly used is using wireless sensor motes, which can be deployed in an indoor environment.

Wireless Sensor Networks (WSNs) [2][3] are emerging as one of the key human monitoring technologies in assistive environment applications. WSNs integrate the capabilities of small nodes into a large distributed network performing the tasks of not only in monitoring surveillance applications, but also in assistive environment.

WSNs are able to integrate sensing, processing, and wireless communication in distributed systems. Through advanced mesh networking protocols, tiny sensor nodes form a new community of connectivity that extends the reach of cyberspace out into the physical world. Placement, coverage, and low sensor power are some of the issues in implementing a WSN in physical environments designed to assist a human in need (assistive environments).

WSNs have been applied in numerous real-world applications, such as surveillance, healthcare, inventory tracking, industry automation, military uses and security. Each node in a WSN has limited capabilities, but when connected as an ad hoc distributed system, it is capable of cooperative processing and communication.

1.2 Motivation

The stimulus behind this research originates from two aspects current in the world today. First, the aging of the population as time has moved on. Secondly, technology is continuously changing and upgrading. The older proportion of the population may be growing larger, but technology has a trend of becoming smaller.

1.2.1 Aging of the Population

As the population is aging, more and more people require additional health care, either at home, in the work place or in a nursing facility. Now, a need exists for health monitoring outside of hospital conditions. These new conditions make this technology of interest for developing health care monitoring systems that can be deployed in many different environments, including the home. Other systems in development employ a wide range of different sensors, including cameras, and recording the information for processing. These systems all involve using an apartment environment seeded with sensors for detecting human behavior and activities. While these systems are embedded in assistive environments, they do not have a comprehensive approach to describe events, or handle a general and rapid deployment into different configurations using wireless technology. In this paper, we are presenting our ongoing project of deploying sensors into an assistive environment. We currently are using SunSPOT sensor motes, where each one has been programmed for a specific role based on rules describing events. In addition, we are developing a voice recognition system for reaction to human input in the same environment. Our system can be rapidly deployed without requiring additional wiring or unwanted intrusion into the human patient’s life

1.2.2 Advancement of Technology

With recent advances in Micro-Electro Mechanical Systems (MEMS) technology, wireless communications and integrated circuit fabrication, Wireless Sensor Networks (WSNs) [2], [4] are an emerging new type of ad-hoc networks. WSNs integrate sensing, processing and wireless communication in distributed systems being used in different fields. Through advanced mesh networking protocols, tiny sensor nodes form a new community of connectivity that extends the reach of cyberspace out into the physical world.

1.3 Problem Definition

With all the different needs that exist within assistive environments, several problems arise that a computational framework would provide the solutions.

1.3.1 Apartment Testbed

First, a testbed of sensors and devices would be put into the model of an apartment, which will be covered in section 2.4. Furniture, walls, and appliances are placed to simulate a living space for experimentation and study. Into this testbed, various sensors need to be placed for event detection for monitoring human activity.

1.3.2 Deployment of Sensors and Devices

Sensors and other devices have to be able to both be placed within the assistive environment as well as being worn on the subject's person. In addition, these sensors not only have to be able to sense the environment, but also be able to communicate their results quickly in order to maintain

1.3.3 Data Source for Other Projects

In order to generate information about human activity, machine learning techniques require a data source for input to their routines. Models, dictionaries, grammars and other approaches all require input to function, both for their learning cases as well as for producing experimental results that can detect human activity.

1.4 Summary

Technology is advancing, allowing more complex and smaller devices to be built for monitoring human activity. This activity could be in the workplace, at home, or on the road, but mainly to observe what a person is doing and to render assistance in a task. New technology such as wireless sensor networks, can be quickly deployed into an environment to detect events, which in turn can be inputs to various algorithms. One area of interest is to generate tools for use with looking towards assisting the elderly and improving their quality of life. To that end, an apartment testbed is the has been filled with sensors and devices for generating data as input sources for multiple projects.

CHAPTER 2

PRELIMINARIES

2.1 Introduction

Before going into details of the tools and methods of the computational framework, some preliminary setup and definitions need to be stated. Thousands of sensors and devices exist, and many different types of events exist. In addition, what is the layout of the testbed and how are all these different devices connected.

2.2 Cyberphysical

Cyberphysical means the combination of the real world and the digital world. In other words, a cyberphysical system can detect changes in the environment it inhabits and to input the resulting information to use via a computer. To do this, cyberphysical systems contain many types of different components working together in order to determine the condition of the environment and the presence of any entities working in that environment.

The interaction between physical and digital is common enough in the modern world. Factory floors, car assembly plants, and other industrial systems are full of robots performing repetitive tasks. Kiosks at airports, ATM machines at banks, and other small systems that interact with the user and the environment to dispense information and currency also interact with the environment. Even the scanner at the grocery store that reads the barcode and inputs the identity of the item is a cyberphysical system.

In the Polymnia cross-media platform, cameras are placed within an amusement park to track the progress of a person through the park and the rides.[5] The goal is to be able to create a day journal for the park visitor to take home, but the system was adapting to the human presence automatically without the presence of a human photographer by employing a grammar-based reference tool to link the various camera angles and viewing positions. [6]

A more practical application of a cyberphysical system to the human presence is an Assistive Cyberphysical System. In these cases, a pervasive environment can be deployed to record human activity and provide a feedback loop to improve the method for event detection. [7]

2.3 Assistive Living

An assistive environment is a living or working space fitted with technology to aid a human in completing some job or task. In the case of assistive living, the technology is deployed to increase the quality of life of a human living within the environment. Sensors can sample the information, and automatic and robotic systems can then act upon the information to aid the human subject. At home or at work, pervasive technologies can be employed not only to assist on the job, but to observe the needs of the human and to try and predict what the person will need. For example, at work on a factory floor or workshop, the exhaustion level of the individual is an important factor. Should mistakes be detected or a variance from a normal situation be spotted, it could simply mean the worker needs to take a break. In other cases, the person could be elderly, and in need of a reminder to undergo therapy or to take their medication. Any environment where the computer helps improve the quality of living is an assistive living environment.

2.4 The Assistive Living Apartment

The setup of the test bed has changed over time as new technology, furniture, and infrastructure has been acquired. The illustrations bellow show the development of the apartment over time as new equipment and furniture have been added.

Phase 1 in Figure 2.1 consisted of two rooms, a student's bedroom with bed and desk, and a den area. Phase 2 in Figure 2.2 added a kitchen space with refrigerator and dining area. Phase 3 in Figure 2.3 added a bathroom area and a television watching area. In the final version Figure 2.4, additional furniture and an electric fire place were added. The apartment continues changing as more experiments and fixtures are added.

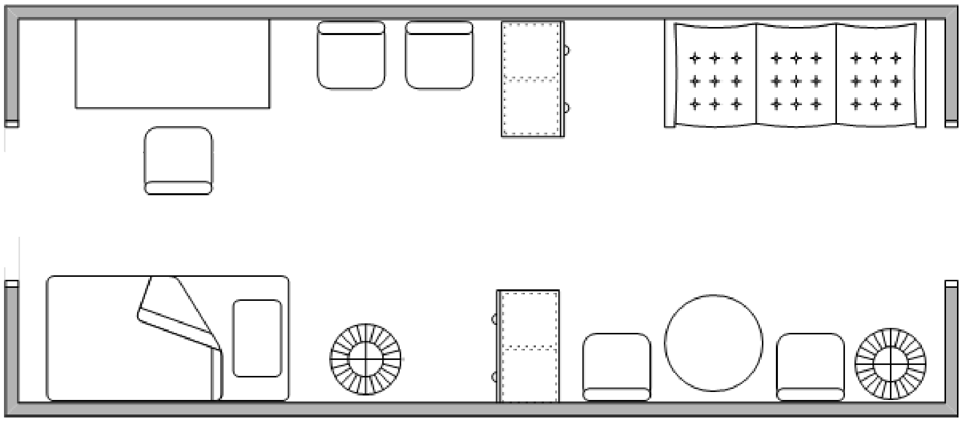


Figure 2.1 Apartment Phase 1 Two Rooms

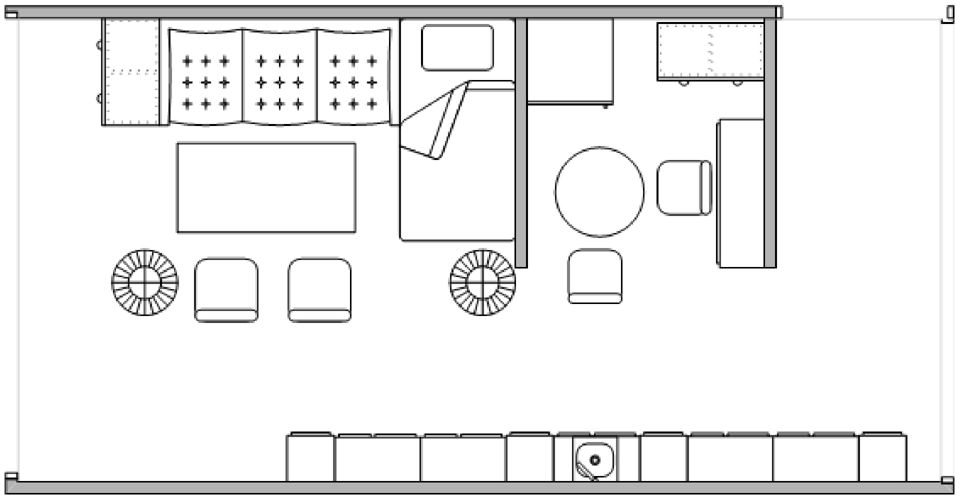


Figure 2.2 Apartment Phase 2 Bed, Hallway, Kitchen

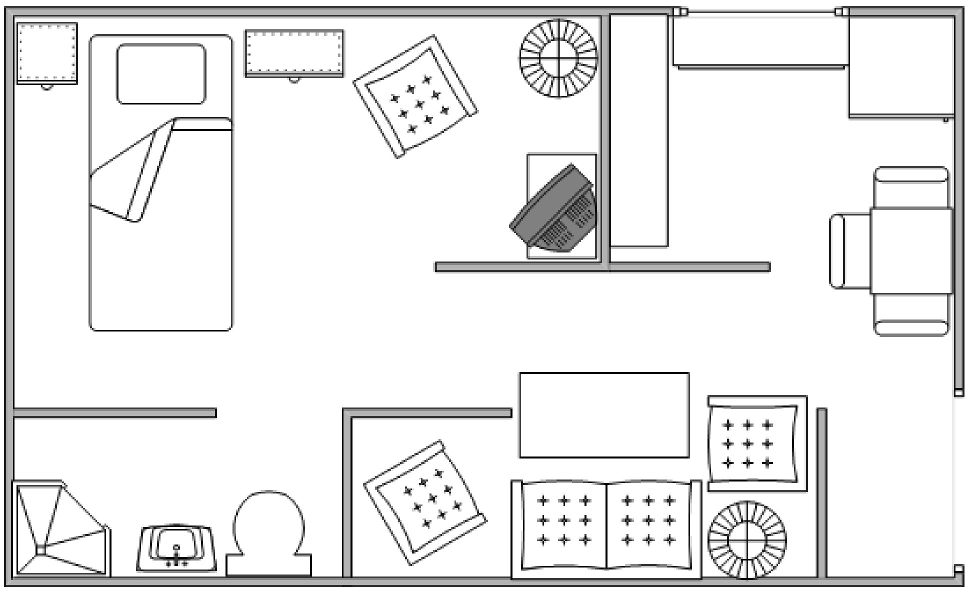


Figure 2.3 Apartment Phase 3 Bed, Bath, Den, Kitchen

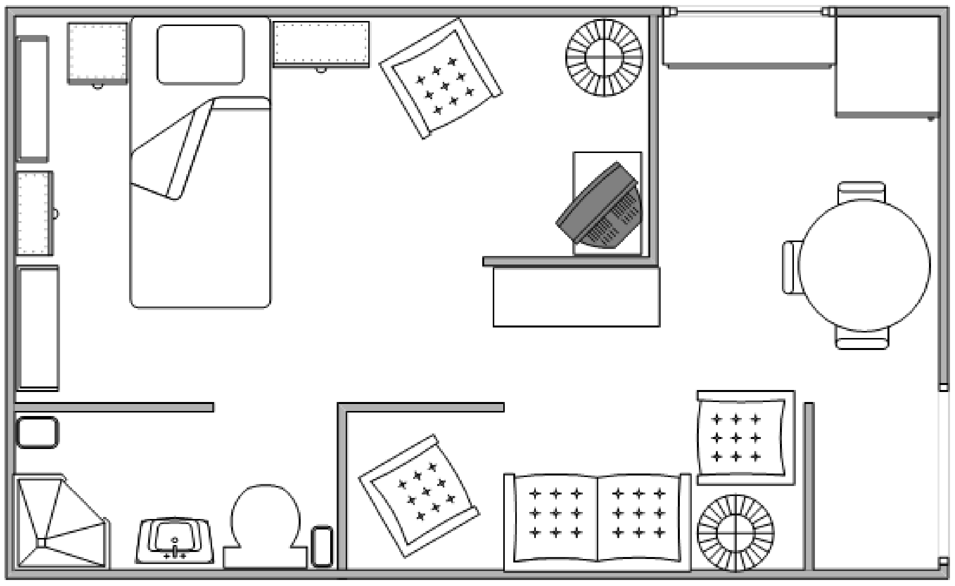


Figure 2.4 Apartment Phase 4 New Dinette and Fireplace

2.5 Technology and Devices

The first stage of development of building a system to recognize human activity within a living space involves getting the technology to function correctly. In order to establish how things work, we first have to take a look at the technology involved in order to proceed with the hierarchal construction of larger systems. This means understanding the sensors involved, the devices that control various sensors, and the architecture of how to combine the various pieces into a working system.

2.5.1 Sensors

To gather information from the environment and into the digital world of computers, the lowest level device is a sensor. A ***sensor***, from the American Dictionary [8] is *a mechanical device sensitive to light, temperature, radiation level, or the like, that transmits a signal to a measuring or control instrument.* At its most basic form, a has the ability to take a reading from the environment. Next, an interface, such as a timer or a A/D convertor, handles the powers and the digitization of the readings. The input for a sensor comes from the environment, and its output is a raw data stream, usually voltages or timing information.. The basic sensor is *not* a computer by itself. Devices can be as simple as a switch, or as complicated as a Microelectromechanical system (MEMs) device. But for these experiments and systems a variety of sensors are incorporated into the various devices that make up the components of a cyberphysical assistive living framework.

2.5.1.1 *Accelerometers*

Accelerometers are sensors that output a reading based upon the amount of acceleration, shock, or vibration. MEMs accelerometers tend to operate on the change in electrical characteristics within a microchip as vibrations impact the sensor. For example, in some MEMs accelerometers, the circuit is built with a moveable plate in a capacitor bridge. As the accelerometer moves, the plate subtly changes its position and the capacitance changes accordingly. [9]

2.5.2 Infrared Range Finder

An infrared range finder consists of an emitter and a receiver that returns a voltage based on the amount of energy received. When not reflected, the range finder generates a low voltage, but when the sensor is blocked by a person or a thing, the infrared light is reflected by the closer object. As a result, the voltage increases and a signal can be triggered. The behavior of the range finder in this case is that of a proximity sensor, detecting objects that reflect back the energy to the unit. [10].

2.5.2.1 *Passive Infrared Detector (Motion Detector)*

Passive Infrared Detectors (PIR) work by having a charge surface covered with a special laminate that reacts to heat. When exposed to infrared radiation, the surface, which can be part of a capacitor in a circuit, will no longer be able to hold a charge.[9] This loss of electrical charge causes a change in the voltage on the circuit and therefore can determine if a warm body, such as a human, is nearby.

2.5.2.2 *Photocell*

Photocells and photosensors refer to a wide range of devices from photodiodes to phototransistors. One type of photocell is a passive photovoltaic cell, a sensor that converts the incoming light into a voltage that can be read and translated into a computer. This sort of sensor is useful for detecting the change in light in an area, whether a lamp has been switched off or if an obstacle, such as a human, has crossed its path.[11]

2.5.2.3 *Temperature Sensors*

Temperature sensors come in various types, from thermocouples to thermistors. In thermocouples, a connection is made between two metals that have different properties. As the temperature increases, so does the physical condition of the bimetallic sensor resulting in a measurable change in voltage. A second method is the use of thermisters, which are an on-chip sensor that can be controlled digitally.[9] Generally, as the temperature changes, the resistance within a circuit on chip also changes, and the resulting change in either voltage or current can be measured.[10] A final type of sensor is the band-gap temperature sensor that is sampled 16 times before and then averaged to reduce noise. [12]

2.5.3 Devices

Sensors by themselves cannot produce the necessary data, rather they are combined with other components to create devices for the recording and exchange of data. To do this, various sensors can be connected to or are a part of different systems. The Phidget is a sensor kit system that provides a USB interface. The RFID reader is a complicated device that can read identity numbers off of tags. The bed pressure pad is actually a grid of 32x32 force sensors connected to an interface. These complex systems are devices.

2.5.3.1 *Phidget Interface Board*

One useful device is the Phidget interface board, in Figure 2.5, which has the capacity to run up to 8 analog sensors, as well as 8 digital inputs and outputs. The interface board creates a simple means for interfacing sensors with any computer that can support the driver and a USB port. This array of channels allows the interface to report on any sensor that has an effective range between 0 and 5 volts. [13]

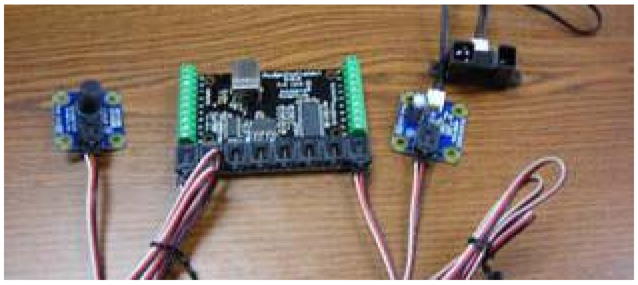


Figure 2.5 Phidget interface board with sensors attached.

2.5.3.2 *Radio Frequency Identification Reader and Tags*

RFID is radio frequency identification, the way of placing a physical tag on an object, and being able to store an identity number than can be read without using a line of sight. [14] RFID is an enabling technology that improves efficiency, prevents errors, saves costs and increases security. In addition, RFID technology is now used in smart packaging that allows the creation of a tool that records when patients take medication, and how much they take. Such a smart tool can also provide prompts to help them comply with the doctor’s instructions.

2.5.3.3 *Pressure Mats*

One method of detecting the passage of a person in the assistive environment is adding a pressure pad to key locations within the living space. These pads, seen in Figure 2.6, respond to the weight placed upon their surface and perform a contact closure to complete the circuit.[15][16]

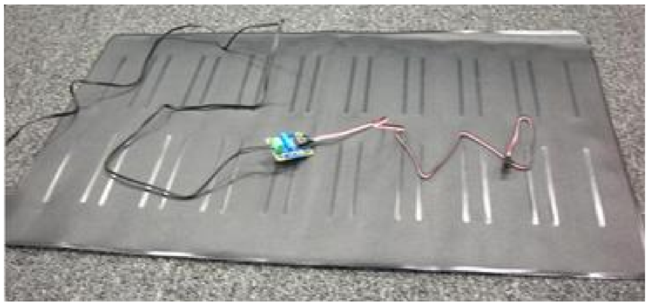


Figure 2.6 Security pressure pad

2.5.3.4 *FSA Bed Pressure Pad*

A further piece of technology incorporated into the apartment setting is a high-resolution pressure pad [17] place under the mattress pad on the bed. This device would record the position of the human body on a rectangle of 32x32 sensors the size of a twin bed and this can generate an image using an off-the-shelf application. For the experiments, the bed pressure reading was taken as well for future analysis.

2.5.4 Motes

A mote, or sensor node is a miniaturized embedded platform that was first described during the Smartdust project in 2001. [18] A mote consists of not only the sensors, but an onboard computer, a power source, a interface board for the sensors, and a radio transceiver to broadcast the results.

2.5.4.1 *T-Mote Invent Wireless Motes*

The T-Mote Invent sensor nodes consisted of a small operating board of the Telos design from Berkley with the addition of its own packaging to include a photocell and two accelerometers. This system was configured using the TinyOs and TinyC operating platform that could be connected to a gateway that could translate from the WSN to ethernet. Small, compact, and operating on the Telos motherboard, these motes had custom cases that included a lens to protect the photocell. [19]

2.5.4.2 *SunSPOT Wireless Mote*

The SunSPOT sensors consist of a small computerized board operating with a 2.4 GHz radio center, as can be seen in Figure 3. Each of these motes has a sensor daughter board with a set of different sensors. Each SunSPOT can recognize the voltage of the battery, the luminance of a photocell, the current air temperature from a Thermistor, and the force affecting each spot in the form of the acceleration measured along the X,Y, and Z axis.[20]

2.5.5 Architecture

The physical setup of an assistive living environment framework consist s of the sensor nodes, additional devices, and of course, the environment itself. The majority of the sensors are deployed on a Wireless Sensor Network (**WSN**) for ease of deployment in living environment. Additional devices are available to supplement the WSN, either medical devices or other sensors.

2.5.5.1 *Wireless Sensor Networks Architecture*

A Wireless Sensor Network (**WSN)** is composed of many components working together in order to gather information and then to relay the result to where it can be used. Wireless Sensor Networks can combine into various types of configurations in order to detect changes in the assistive environment. In casual conversations, a unit deployed in the mesh will be referred to as a sensor, when actuality, the device involved is much more. The various components in a WSN include the sensor motes, base stations and gateways, and then external entities including servers, databases, and even web applications. The general setup of this framework can be seen in Figure 2.7.

A **sensor node** or **mote computer:** Give a set of sensors power to run, computational power, a transmitter to report their results, and an operating system to run on, and we will find a device called a sensor node or mote computer. These devices are referred to as nodes, motes, or even as sensors. These synonyms can be very confusing, and for the rest of this discussion these devices will be referred to as ***nodes***.

A **base station** or **sink** is a node in the WSN that is the receiver of the data from the various nodes. Through either direct broadcast or from a multi-hop link, the base station will receive the sensor readings and forward them on to another device. These devices could be the host machine or a gateway.

A **gateway** is any device that can make messages cross media boundaries. In order to access the WSN, we must first gain access to a system capable of moving the data from one hardware specification to another. For example, a Bluetooth sensor device is connected to a desktop PC. The Desktop PC is then connected to a server via a network run on Ethernet cable. The desktop PC acts as the gateway for the messages. Additionally, many systems have a special embedded computing device whose sole function is to relay information between the mediums, or through a secondary mesh at the gate way level. [21] For example, we cannot hack into a WSN via the internet from another city, without some device that can translate our commands and broadcast them to the WSN.

Servers and Databases: After going through the WSN, the BaseStation, the Gateways, and various host machines, the data and messages have to be put somewhere. If the data is to go to the internet, then a server would be set up to relay the information to a web host. Alternatively, the server could have a database running on the backend, saving the messages for later study.

A **host machine** is usually a computer that has been set up with special software and drivers to be able to program individual nodes in a WSN setup. This machine usually also can host a node to behave as a transceiver for communicating with the network. [22]

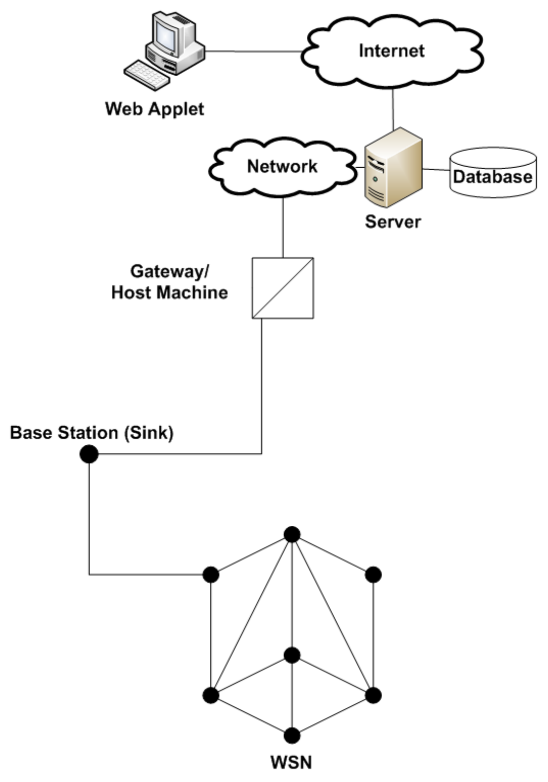


Figure 2.7 Architecture of a WSN for Assistive Living Environment

2.6 Types of Events

Everyone has their own definition of an *event*, which can be very confusing. Events occur at different spatial-temporal granularities and in different domains. At its basic form an event is some reaction that occurs at some given time, whether it be the click of the mouse or someone falling in an assistive environment. With so many different types of events, it is useful to define two specific types of events for the purposes of discussion: ***episodic events*** and ***events of interest***.

2.6.1 Episodic Events

When sensors nodes are placed in an assistive environment, the onboard sensors will start continuously sampling from the environment when they are in data collection mode. Given that the sampling rate being used is around 66 Hz, that means in one minute, 3960 samples are taken. If we assume five sensors are active on the mote at that time, each mote then has sampled 19800 times. Now, extending the sensor to a mesh of twenty motes, that brings the amount of data sampled up to 396,000 samples taken in one minute. To send this much information back to the base station on a single channel is unreasonable. A bottleneck will always occur when the bandwidth is swamped with so much data on such a network. Instead, the messages are going to be broadcast when an *episodic event* occurs.[23][24]

An *episodic event* occurs when the conditions in the environment meet a preset set of parameters programmed into each sensor node. Only at these times will the mote broadcast the sensor input back to the base station, keeping the bandwidth usage of the WSN low. As we will discuss in section 2.4, different interactions between the human and the sensor nodes placed within the apartment will generate a different type of episodic event. These episodic events can then be used as a source for further analysis. And *episodic event* is a low-level event that occurs on a sensor node. Raw data is to be processed on the mote itself, and the resulting event is then broadcast over the WSN. For more details, see section 3.3.

2.6.2 Events of Interest

One concept to be defined are the *events of interest.* Possibly thousands of combinations of episodic events exist within any assistive living environment, and their combination and sequencing can lead to the detection of many types of human activity. Certain sequences of events have some risk associated with them, and become events of interest as seen in Figure 2.8 through Figure 2.12.

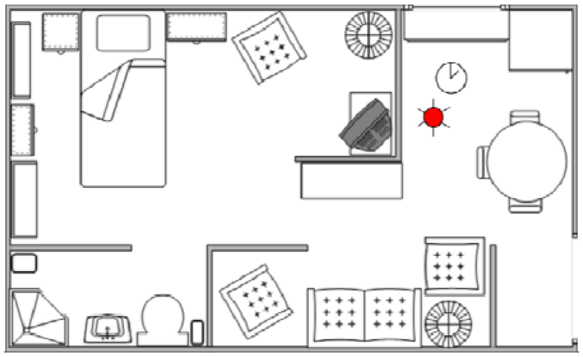


Figure 2.8 Loss of Signal: Subject Fails to Move After Some Time

Loss of Signal-The subject in the assistive living environment suddenly stops being detected in the apartment.(Figure 2.8) For some reason, the various sensors, both pervasive and worn, are no longer able to track the individual. Given these conditions, a risk exists that the person has stopped moving for an unknown reason, or could be in danger.

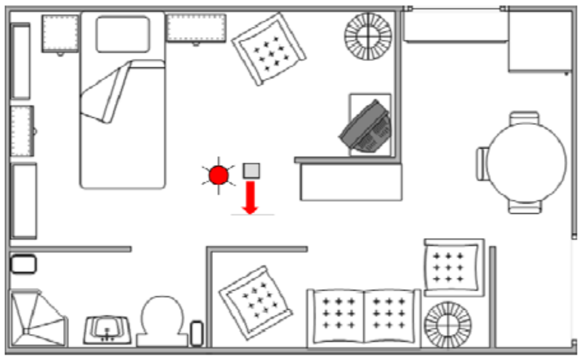


Figure 2.9 Falling Down

Fall Detection- Various types of sensors can be used to detect an impact within an assistive living apartment.(Figure 2.9) Projects have used microphones in the past [25] or can be done using a worn sensor [23][27]. Any time a person in need of assistive care suffers an unexpected fall would be an event of interest.

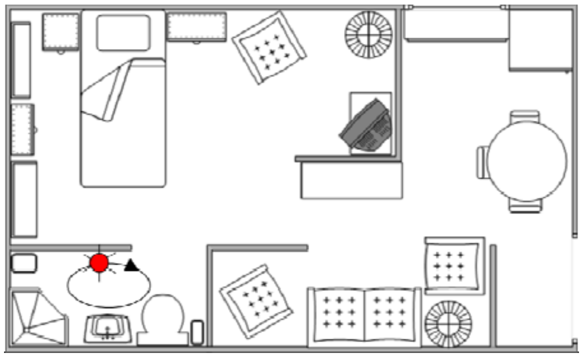


Figure 2.10 Entrapment: Enters an Enclosed Space and Never Leaves

Entrapment- One event of interest describes the case of an individual entering a bathroom or closet or other confined space and not leaving. (Figure 2.10) They could be mobile, not having a fall, or other event of interest. They instead spend a longer than expected period of time closeted in a small space. Spending several hours within a bathroom or a closet would be an event of interest.

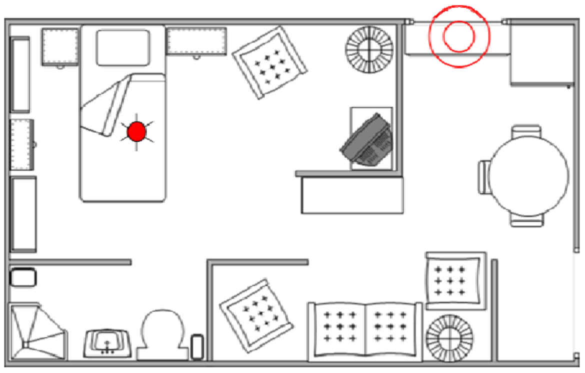


Figure 2.11 Appliance: Leaves an Appliance On

Appliance-One worry is the case of a human turning on a cooking stove, an oven, or even leaving a television on, and then wandering off to take a nap either in the bedroom or on a sofa or on a chair.(Figure 2.11) Some device has been switched on and then the user no longer is in motion, or has even been detected getting into bed. This set of conditions would be alarming to a human, and if detected, intervention should be necessary.

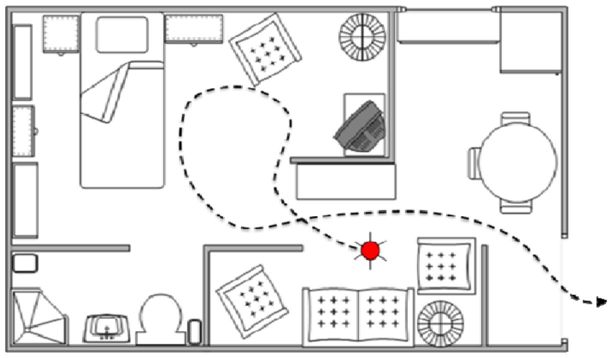


Figure 2.12 Sudden Departure: Leaves Areas Unexpectedly

Sudden Departure-One worrying case for assistive living is when the subject is suffering from Alzheimer's disease.(Figure 2.12) A recurring case is the flight of the patient. Usually at night, perhaps when the subject is confused and in pain, they will simply flee, trying to escape their own reality on foot. So, an event of interest would include the sudden departure from the apartment, perhaps even tying into a daily planner system to see if the exit was expected.

2.7 Summary

Before going forward with the investigation and construction of an apartment testbed, it was first necessary to look at what kinds of sensors, devices, and sensor nodes that are available for use. Different components have been defined that, when assembled, will become the testbed environment for observing human activity. The main backbone of the system is a Wireless Sensor Network, which can be configured with architecture to deploy sensor nodes quickly. In addition, key events, episodic events and events of interest, have been defined. The next stage then is to start investigating the pieces of the apartment, both physical and digital, to develop the testbed.

CHAPTER 3

THE CYBERPHSYICAL FRAMEWORK AND THE ASSISTIVE LIVING APARTMENT

As discussed in Chapter 1, a simulated apartment was used as the focal point of this research. The configuration of this apartment has changed over time with the acquisition of technology and furniture, with different problems being raised and solutions found. To that end, several sets of experiments are detailed below. These include the automatic placement of sensors within the apartment, a study of WSN architecture and sensor nodes, and the development of episodic events for the detection of human activity, both with sensors placed in the environment or worn by a test subject.

3.1 Automatic Placement of Sensors

In this section is a description of a method for automatic sensor placement. The assistive living environment can be modeled as a 3 dimensional space, and it is possible to create a method to automatically place models of the sensors within that volume. This method first requires the construction of a ***volume of inter*** to represent the target space. A ***sensor model*** is then defined in general terms using geometric constraints. Next, a ***voting scheme*** is used to accumulate evidence to find the best sensor poses to observe the volume of interest. Based on this evidence, a greedy heuristic selects a set of sensors to find the ***coverage*** of the target space. All these pieces have been implemented, which was used on both synthetic and real world cases. One of these real-world cases was the Heracleia Assistive Living Apartment.

3.1.1 Volume of Interest

The *volume of interest* describes the portion of the three dimensional model of the assistive environment where sensor coverage is desired. Usually, this includes the volume in which a human being would be walking or resting. First, some assumptions must be made about the target space to be observed. Human beings are usually less than six feet high and tend to walk on the top surface of structures such as the ground floor. Therefore, the volume of interest is constructed from any polygons that have a near-vertical normal vector.

A near-vertical normal is a normal vector within ε degrees of the vertical axis, where ε is a constant that represents an upper bound for the amount of slope a structure may have in order to support a human subject. Assuming the 3D environment is composed of a triangular mesh, each triangle can be considered to be one face in the model. The result is that the model has set of faces *F* and vertices *V*, where each face *f* in *F* is defined by 3 vertices (V_1, V_2, V_3). The normal vector *n* of each face *f* can be determined from these three points.

With the normal known, we verify if the angle between the vertical component (*z*-axis) and the normal vector is within ε degrees. The vertical magnitude is compared with the horizontal magnitude to determine the angle between the normal vector and the tangent. Once this angle θ is found, if the normal vector *n* is within +/- 4.5 degrees of a 90 degree angle, then the surface is defined to be horizontal and a human could walk on the surface. This property makes that particular face to be used to generate part of the volume of interest. If a normal is pointing along the *z* axis, then the corresponding face can be considered to be horizontal and, consequently, this face is in the support structure (*i e* , floor) of the model This can be seen in Figure 3.1, where the magnitude of the vertical component of the normal vector is compared to the magnitude of the horizontal component to determine the angle of the plane.

$$r_1 = \overrightarrow{V_1V_2},\ r_2 = \overrightarrow{V_1V_3},\ n = r_1 \times r_2$$

$$\theta = \tan^{-1}\frac{\sqrt{n_x^2+n_y^2}}{n_z}$$

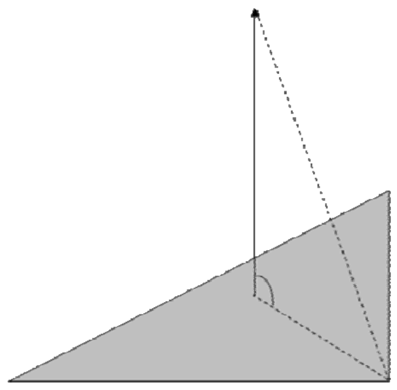


Figure 3.1 Finding the floor faces

Now the horizontal faces have been identified, a discrete grid is built within the limits of these boundary points that make up the triangular face. Next, each point on the grid must be verified to be within the triangle of a floor face In Figure 3.2, points in blue are within the triangular face and points in red are outside the face.

Let A, B, and C be the vertexes of the triangle.

Let P be the point to be investigated.

, ,

,

If d is greater or equal to zero, then the point lies between the two vectors. And if, in turn, the point lies between the two vectors of each of the three corners of the triangle, then the point can be said to be within the boundary of the triangle.

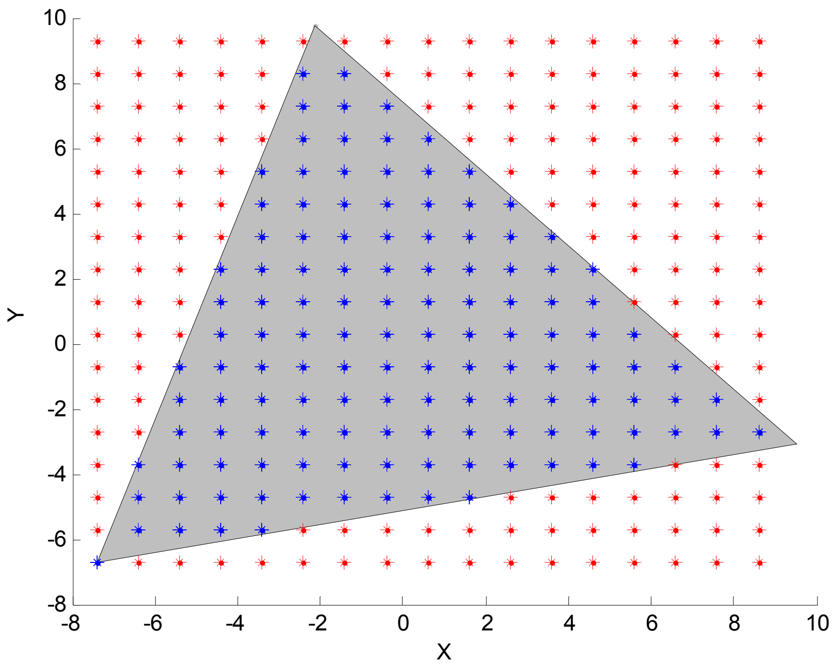


Figure 3.2 Valid points within a face.

Now the points that lie within the face are identified, the volume of interest can be generated. The volume of interest extends these support polygons to a prism that is six feet in height above a floor face as can be seen in Figure 3.3. This is the target space to be surveyed for human activity. Once all floor faces have been identified, then the points in these faces with minimum and maximum coordinates of the *x*-axis and *y* axis can be determined

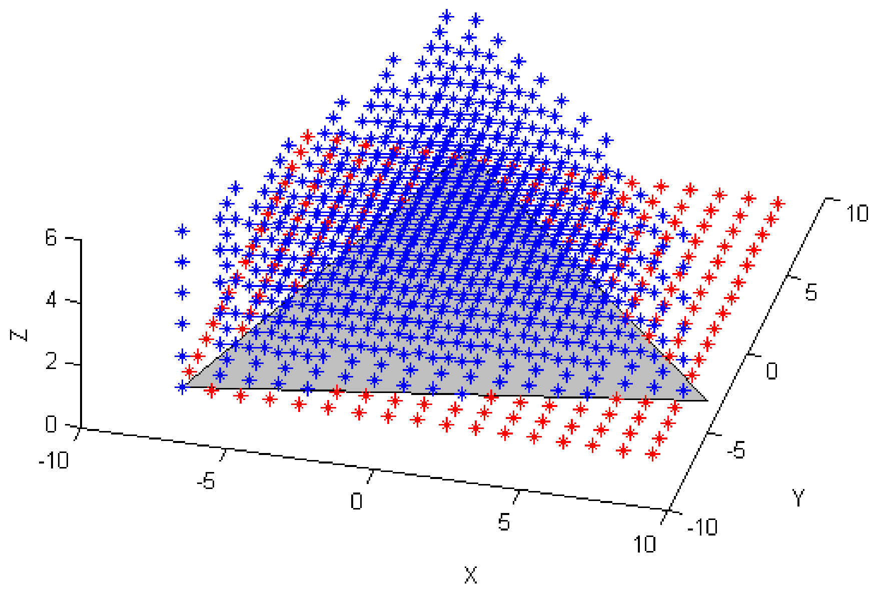


Figure 3.3 Extending valid points in the z-axis.

3.1.2 The Sensor Model

The ***sensor model*** is represented by the three-dimensional solid to represent the space perceptible to the given sensor. For example, an orthogonal camera model can be represented as an orthogonal box. In the following experiments, the sensor is treated as an orthogonal camera with dimensions of 4m by 1. 5m by 1.5m for the length, width, and height of its bounding box. The volume that the model camera would be able to see is defined by this set of ranges. Many other camera models could be created with this geometric method. A projective camera is modeled as a frustum solid. A stereo pair of cameras may be represented by a pair of frustum solids. An omni-directional microphone or a radio transmission could be modeled as a sphere. Each sensor model has a fixed pose (location and orientation) from where it can perceive the assistive environment. The location corresponds to the position of where the sensor would be mounted. This center could be the center of projection for a camera for example. The sensor's location can be constrained to be at one of the points in the set generated for the volume of interest. The orientation is represented as the rotation that aligns the local coordinated system of the sensor with the global coordinated system of the environment. The location and orientation of a sensor compose its pose and fully specify its geometric configuration in the 3D space. The sensor's pose is the parametric space used to select a set of sensors and their corresponding poses in the environment to cover the volume of interest. In Figure 3.4, the camera is modeled as a box with an anchor point marked in blue

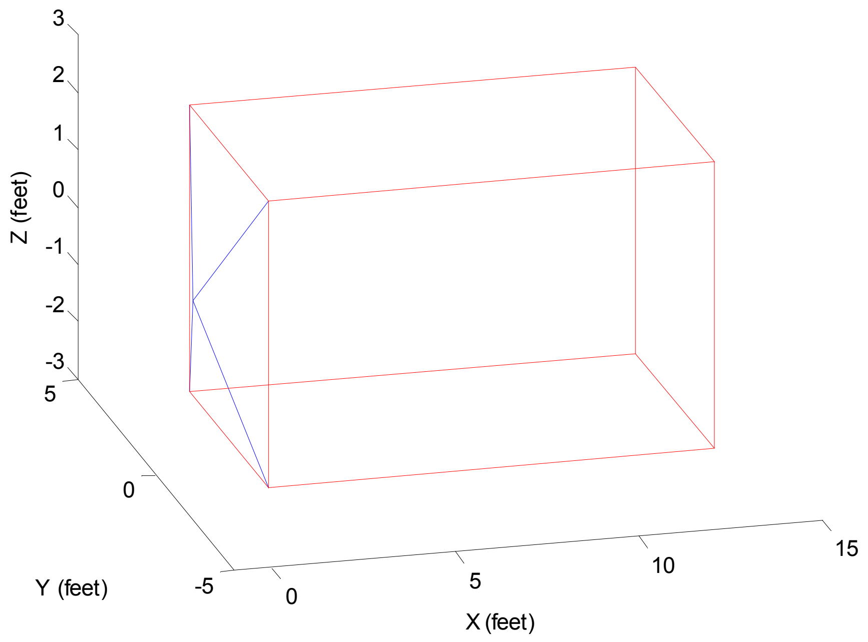


Figure 3.4 An orthogonal camera model.

3.1.3 Voting Scheme

The goal was to find out how many sensor models are needed to cover the maximum amount of space within the volume of interest To do this, a voting scheme had to be created that would assign a voxel in the volume of interest to each one of the sensor models generated for the experiment. Each sensor model was placed upon a point in the volume of interest, and rotated on all three axes, and then at each of these possible positions and orientations (six degrees of freedom), each voxel in the volume of interest is checked if it is perceptible (*e. g.* visible, audible) by the sensor model. If the voxel can be perceived, then that instance of the sensor model receives one vote for covering the volume at this particular location and orientation. In order for a voxel to be perceptible by the sensor model, however, it is necessary that the voxel fulfills certain requirements. The voxel had to be within the viewing range of the camera If the voxel is further away than the 4m limit, then it will be ignored and will not be counted.

An additional constraint that can be applied to sensors is if a voxel is blocked or occluded. Cameras cannot see through walls, and radios cannot transmit through metal. Using the camera approach, no other face can lie between the segment defined by the camera's center of projection and the voxel being tested. With these endpoints known, it is possible to find the point where the ray intersects each face's plane.

That intersection point can then be determined to be inside the face defined by the three corresponding vertices If such a point is on the face's plane, and lies within the face, then the view of the camera model is blocked. If a camera position is at a point *p* and the current voxel of interest is *v*, any face in the model had to be checked to see if it lies on the ray from *p* to *v*. Next, the routine finds the plane for each face in the model and intersects the ray *pv* with each plane. If the intersection point lies on the surface of the plane and is inside the boundary of the face, then the line of sight is blocked and no vote is given for the election of that camera for that voxel of interest.

A plane ***p*** can be defined using three points, $\boldsymbol{V_1}$**,**$\boldsymbol{V_2}$, and $\boldsymbol{V_3}$**.**

Two vectors can be defined as

$$r_1 = \overrightarrow{V_2V_1},\, r_2 = \overrightarrow{V_3V_1}$$

And the normal ***n*** is the cross product of these two vectors

$$n = r_1 \times r_2$$

From this, the coefficients for the plane can be defined as

$$a = n.x, b = n.y, c = n.z$$

$$d = -1(aV_1.x + bV_1.y + cV_1.z)$$

And the general formula for the plane is:

$$0 = ax + by + cz + d$$

Next the line-of-sight for the sensor between the mounting location $\boldsymbol{p_1}$ and the target voxel $\boldsymbol{p_2}$ can be defined as

$$u = \overrightarrow{p_2 p_1}$$

Now, the point on the surface of the plane where the line intersects can be calculated. First, a line *w* must be taken between a point on the line *u* and one of the vertexes V_1, of the face,

$$w = \overrightarrow{p_2 V_1}$$

Next, the distance along the line *u* what is proportional to the line *w* by solving for magnitude *s* with respect to the normal of the plane to find point p_j.

$$s = \frac{n \cdot u}{n \cdot w}$$

$$pj = su + p_1$$

Now, the plane has been found to intersect the line at point pj. The question is does this plane now lie between the two voxels of interest in the geometry. It is quite possible to project the line onto a plane that is on either side of the line segment. To solve this problem, the distance of the point on the plain is compared to the distance to each of the end points.

$$|\overrightarrow{p_1 p_2}| = |\overrightarrow{p_1 p_j}| + |\overrightarrow{p_2 p_j}|$$

Once these conditions above are met, the voxels within the volume of interest can be translated and rotated to be on the same axis as a canonical sensor model. Once a voxel is in the local coordinated system of the sensor, the orthogonal box that defines the sensor model can be used to check if the voxel lies within the sensor model. In the positive case, that sensor model's pose receives one vote for being able to perceive that voxel. Once the number of votes for each possible sensor model position is found considering all voxels in the volume of interest, then find the minimum number of sensor models to cover the entire volume of interest. This problem is solved with a greedy heuristic that sorts the sensors by the most number of perceptible voxels. The sensor with the most number of voxels is included in the final set of sensor positions. After sorting, the volume is updated by removing the voxels already perceived by each sensor model in the final set of sensors, and then again resort the list for the most number of perceptible points until the volume is covered. The coverage for the voting scheme is also constrained by setting an upper limit less than one hundred percent.

3.1.4 Implementation

The main implementation of this voting scheme was done using C++ and a series of objects to represent the geographic components: vertices, voxels, planes, lines, sensor models. The input to the system, however, came from either user defined cases built either from test cases or from real life cases. To that end, two software packages were used to create and translate the various environment models to be used as input.

Google Sketchup is a free, downloadable tool that is quick and useful for generating basic 3D models. The interface includes a properties box that allows the user to quickly specify the dimensions of each polygon or solid added to the model in the 3D World.[29] The product, however, has a very low capability to generate models that are compatible with other systems or provide data in an easy to read format. [29][30]

The Sketchup files, however, can be exported and extracted to generate a Collada style model file. This Collada model in turn can be loaded using the Blender graphical suite, which includes an output file convertor and resizing tools. Combining these programs and formats together, Blender can convert a Google Sketchup model into a Wavefront text file. The Wavefront text file converts all the generated 3D components into a sets of triangular faces and vertices suitable for the voting scheme. As can be seen in Table 3.1 Example of a converted OBJ File, the resulting data file marks out each vertex with a row starting with 'v', and each face in the file is a combination of these vertexes starting with the letter 'f'.

Table 3.1 Example of a converted OBJ File

Blender4D v247 OBJ File: partition.blend
www.blender3d.org
v 0.000000 120.000000 0.000000
v 0.000000 0.000000 0.000000
v 0.000000 120.000000 120.000000
v 120.000000 120.000000 0.000000
v 240.000000 120.000000 0.000000
v 240.000000 0.000000 0.000000
v 240.000000 0.000000 120.000000
v 120.000000 0.000000 0.000000
v 0.000000 0.000000 120.000000
v 120.000000 0.000000 120.000000
v 120.000000 120.000000 120.000000
v 240.000000 120.000000 120.000000
v 19.999201 9.999600 0.000000
v 9.999600 0.000000 0.000000
v 9.999600 9.999600 0.000000
v 19.999201 0.000000 0.000000
v 19.999201 0.000000 9.999600
v 19.999201 9.999600 9.999600
v 9.999600 9.999600 9.999600
v 9.999600 0.000000 9.999600
v 0.000000 0.000000 0.000000
v 0.000000 0.000000 9.999600
v 0.000000 9.999600 9.999600
v 0.000000 9.999600 0.000000
usemtl FrontColorNoCullingID
s off
f 15 13 14
f 14 13 16
f 16 13 17
f 17 13 18
f 18 13 19
f 19 13 15
f 19 20 15
f 14 15 20
f 14 20 21
f 22 21 20
f 22 23 21
f 24 21 23
f 24 23 15
f 19 15 23
f 24 15 21
f 14 21 15
f 20 14 17
f 16 17 14

3.1.5 Experiments

In order to validate the voting scheme method for placing the sensors, the method was first tired on small models of a single room or a single shape. Then, as the system progressed, more complicated combinations including openings and obstacles were added. Finally, the voting scheme was tried on two real world cases, an assistive living apartment and a house.

3.1.5.1 *Experiments with Synthetic Cases*

To begin with, the voting scheme was first applied to a set of synthetic test cases to see how they would respond to the program. More specifically, the synthetic cases are rooms with a triangular, squared, and circular shape. The approach finds the best possible places to put sensors within a volume of interest and, as such, each different shape of a synthetic room corresponds to a different volume to test the placement technique. Depending on the shape of the room and on the openings in the walls, the sensors will be placed to cover the most number of points in the volume of interest.

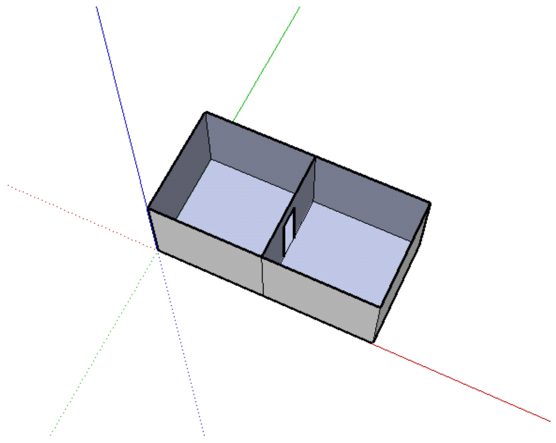


Figure 3.5 Two square rooms connected by a doorway.

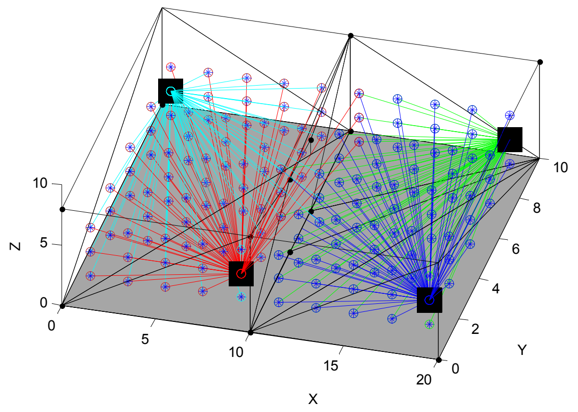


Figure 3.6 Placement in two square rooms connected by a doorway

In the two rooms connected by a doorway case, shown in Figure 3.5 and Figure 3.6, the sensors have been automatically assigned to the corners of the rooms. However, instead of being placed in the lower opposite corners symmetrically, one sensor has switched sides in one of the rooms. This way, the line-of-sight of this sensor can look through the doorway and take votes from voxels on the other side of the inner wall.

This approach also tackles situations where obstacles are placed in a room, as shown in

Figure 3.7 and Figure 3.8, which shows a round floor where a rectangular pillar was put offset on the surface. The method successfully positions the cameras to look around the obstacle. In this case, a rectangular pillar was placed upon a flat circular surface. The voting scheme automatically finds how sensors could be placed in the environment to see as many points in the volume of interest as possible. As a result, the sensor placement forms a partial circle around the obstacle in order to see all the voxels blocked by the pillar.

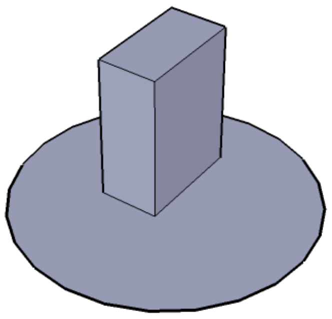


Figure 3.7 Offset pillar case

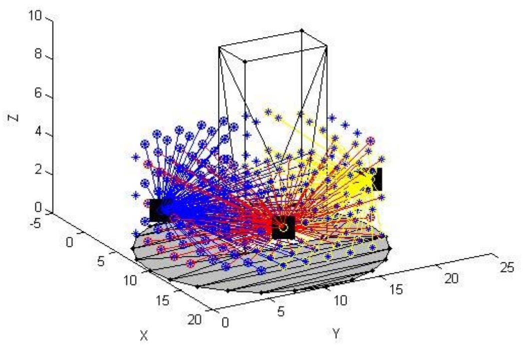


Figure 3.8 Placement in the offset pillar case.

With the synthetic cases, the model of a room is automatically populated with sensors within the volume of interest in order to capture as many voxels as possible. This experiment shows that sensor locations are selected to fill up the entire volume and to see around placed obstacles.

3.1.5.2 *Experiments with Real World Cases*

Two real-world environments were selected to evaluate this approach. In each real world case, the environment was measured and then modeled in three dimensions to generate a 3D model. The 3D models are used as input to place sensors into the environments.

The first real-world environment used the assistive living apartment set up in the Heracleia laboratory. This simulated living space has an area of fourteen feet by twenty four feet organized with partitions to mark off rooms. In addition, the Heracleia apartment is populated with typical furniture for assistive living experiments. For research purposes, the Heracleia Human Centered Computing lab has an apartment layout containing obstacles such as furniture and appliances. For the 3D model, not only were the partitions included into the layout, but the furniture was placed as obstacles within the space for testing our camera placement algorithm. The resulting camera placement configuration, as can be seen in Figure 3.9, places the first camera in the den or living area over the sofa, and looking out towards the kitchen and into the bedroom. The algorithm basically places cameras in strategic places where visibility is maximized.

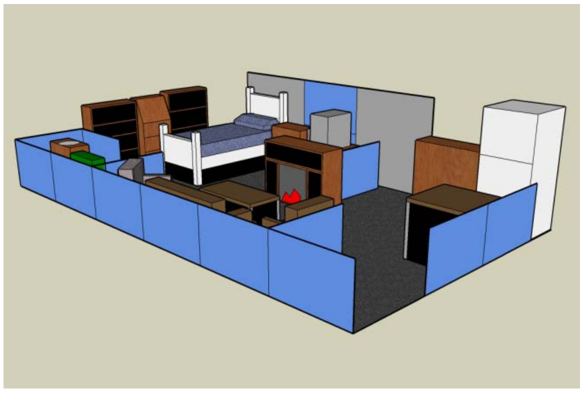


Figure 3.9 Model of the Heracleia apartment.

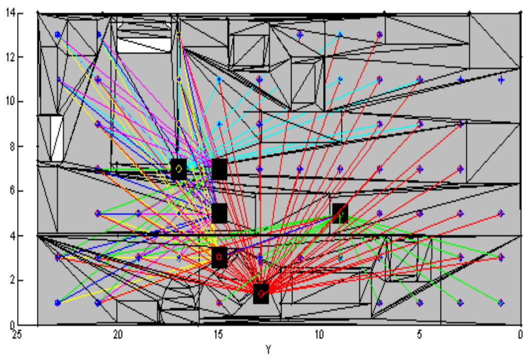


Figure 3.10 Placements in the model apartment.

The second real-world environment is a single family house named Dreamhouse (see Figure 3.11), based on a real, two–story house that was measured and turned into a three-dimensional model.. The corresponding 3-D model is based on the actual dimensions of a real home. The two-story building was painstakingly measured for the height of each wall, the position of each door and window, the location of the ceilings and floor of the second story. The Dreamhouse model was then represented as a mesh, a set of vertices and faces, to be processed. The voting scheme scans the house and generates the voxels for the volume of interest as before. Then, it selects the sensor models. For example, several of the sensors are placed in the vertical shaft containing the stairwell on the Dreamhouse.

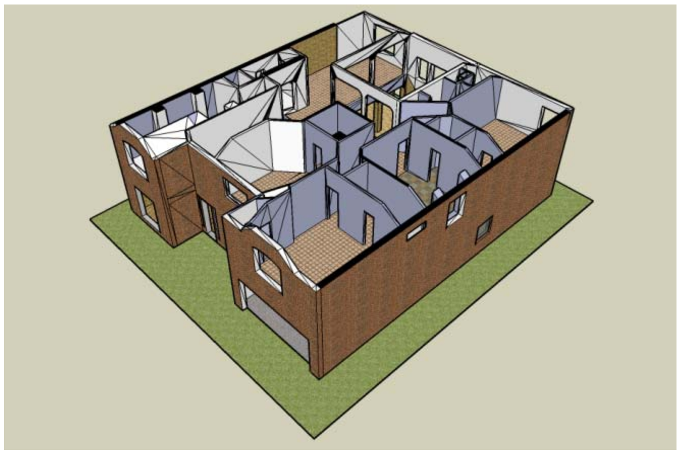


Figure 3.11 The Dreamhouse Model

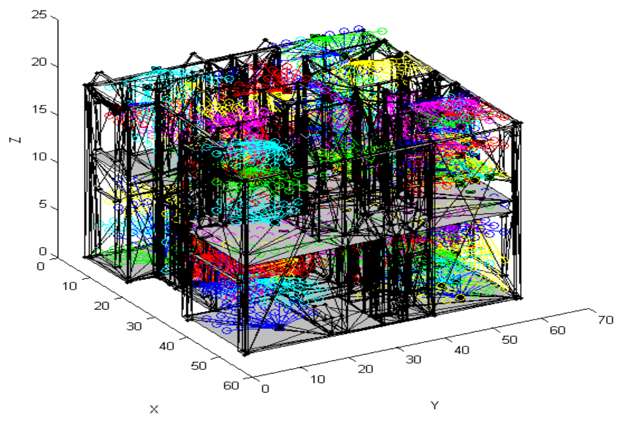


Figure 3.12 Placements in the Dreamhouse Model

3.1.6 Coverage

In order to conclude the processing of the greedy heuristic, a record is kept of how much percentage of the target volume of interest is seen by all currently known cameras in the final set of sensors. After all the sensor models are elected, they are sorted according to the number of votes each model received, *i.e.* the number of voxels each sensor can see.

The sensor model with the highest number of votes is selected first. After that, the voxels seen by this model are removed from the volume of interest, the number of uncovered voxels seen by each sensor is updated, and the cameras are resorted. Then, the next sensor model is taken in turn until the coverage of the volume of interest is greater than some threshold. The threshold used in our experiments ranges from 90% to 99% of the volume of interest. Note that the gain of coverage for each additional sensor (as the number of models increases) is monotonically decreasing, as can be seen in Figure 3.13 through Figure 3.16 .

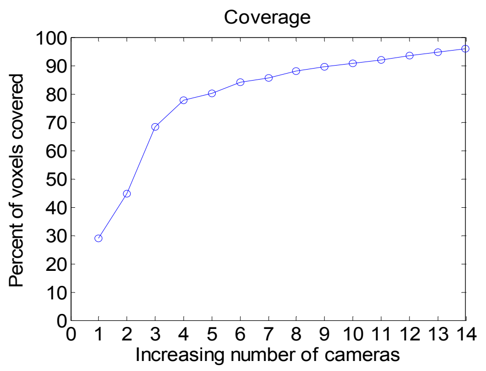


Figure 3.13 Two Rooms and a doorway

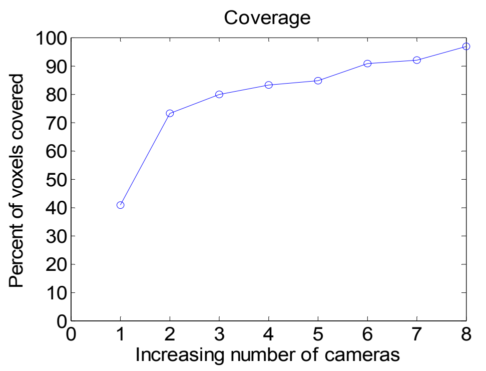


Figure 3.14 Pillar

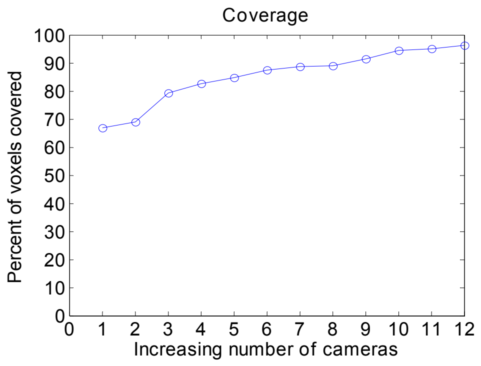


Figure 3.15 Heracleia Apartment.

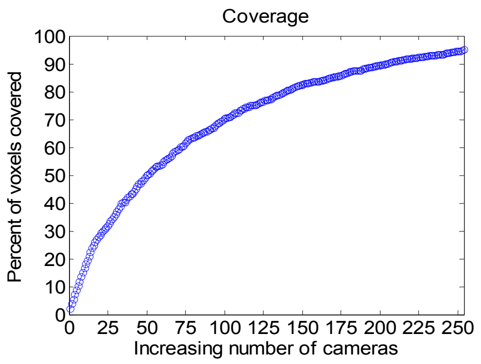


Figure 3.16 Dreamhouse

An additional synthetic case is when there are two rooms completely cut off from each other. where each interior volume is filled separately. In this case, as shown in Figure 3.17, the coverage increases on a distinct pattern for each of the two enclosed rooms.

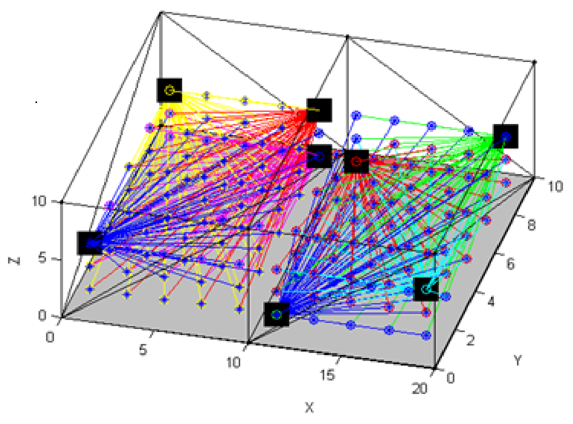


Figure 3.17 Two rooms with no connecting doorway

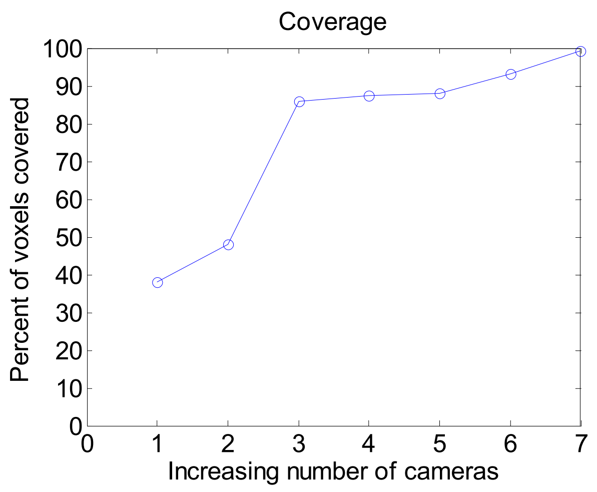


Figure 3.18 Coverage of two rooms with no doorway.

This configuration of rooms shows a repeating pattern, which seems to be echoed in multi-room situations like the Dreamhouse. The subsections have the same diminishing curve and the overall coverage has a similar curve. But it should be noted that the pattern of sensors is different for each room. There are two reasons for this. Firstly, it is possible for two sensors to have a tie in the system. Secondly, the coverage is calculated for the entire model, not just each room separately. An additional sensor can be added to make up the last percentage point of the coverage.

The placement of the sensors is not the same as if they were placed ad-hoc within the environment. For example, if the rooms were a cube, then sensors could be placed exactly adjacent to each other and have four models to cover the volume. The voting scheme, however, separates out the cameras into the corners of the room. Additionally, in the one experiment where the doorway between two symmetrical rooms is offset, the voting scheme placed all the sensors against one wall of the two rooms, so that they could see through the doorway and towards the opposite wall as seen in Figure 3.19. In this case with the doorway being offset from the center of the diagram, all the cameras are automatically placed on the wall with the line-of-sight passing through the opening.

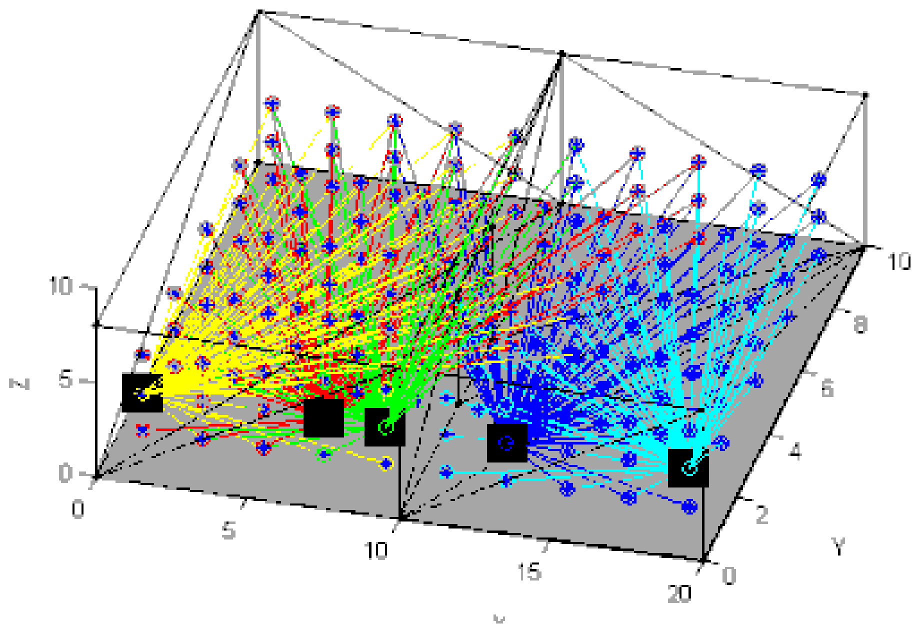


Figure 3.19 Two rooms with an offset doorway.

3.1.7 Summary

A method has been implemented for finding the automatic placement sensors, such as wireless sensor nodes or cameras, within an assistive environment. A voting scheme has been created that uses a 3D model of the environment and a 3D model of the sensor space to find the placement within the environment. Then, the system sorts the sensor models by the number of voxels covered and adds these to a final tally.

3.2 Data Collection

In order to be able to use the sensor nodes in an assistive environment, they first need to be studied to see how they respond. The first family of sensor nodes used for this research was set up in different experiments to observe how they react to changes. The TMote Invent units were used for this preliminary work, and were studied in three projects to look at how the system could be constructed and what approach should be taken.

3.2.1 TMote Invent

The TMote Invent was a sensor node based on the TinyOS platform and the TinyC programming language. Each sensor node was equipped with a microphone, speaker, 2-axis accelerometer, temperature and light sensor. The light sensor in turn was shielded with a funnel-like shade and covering lens. In this setup, any TMote invent could operate as the base station or "sink" for the wireless sensor networks, and relay any data packets over a USB cable. An accessory to this system was a 'gateway" device that allowed the USB port on the gateway to be mapped to a TCP/IP socket. This way, the sensor readings from the nodes of the WSN could always be transmitted to any program with a socket connection. Each TMote could be plugged into a host machine for programming using a USB connector which would also recharge the battery. The TMote set up as a basestation (sink) and the nodes themselves can be seen in Figure 3.20 and Figure 3.21. [31][32][33]

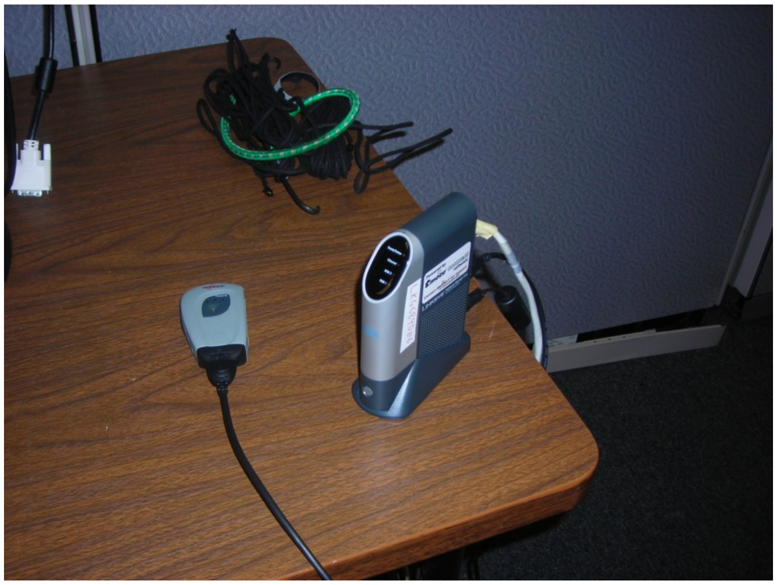


Figure 3.20 WSN Sink and gateway

Figure 3.21 T-Mote invent Sensor Motes

3.2.2 The Three Room Puzzle: TARGET

TARGET had three reserved areas set up, such as cubicles or changing rooms. Non-photographic sensors (i.e. no cameras) would be desired in such locations. The problem is how a responsible party would know if someone had entered into a reserved space that was already occupied. In an assistive environment, a subject would not want cameras in their bedroom or bathroom. So, these less invasive sensors were a better solution. Three areas were set up to be sixteen inches on each side. On three sides, a mote was set up with an active photocell to watch the space. Then, a paper cup simulating a subject would be placed within the target area as shown in Figure 3.22. The goal is to be able to observe the response of the sensors to the combination of targets. As more targets were placed into the zone, the change on the sensors would be recorded.

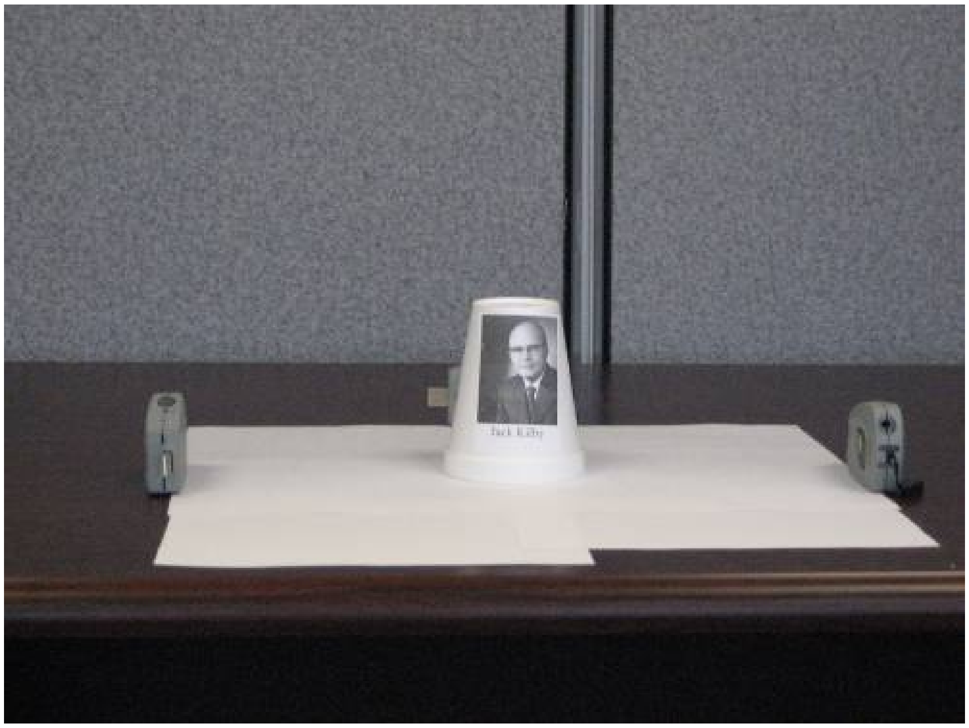


Figure 3.22 System Test Layout.

An applet was devised that would display the motes in the three areas as seen in Figure 3.23. In this figure, two sensors in the center room have detected a change in the environment. The applet responds to the change in the environment in two ways, by changing color and sounding a chime. When the sensor increases in voltage due to the presence of more light because an object has been moved away, the sensor display will turn red. When an object has been moved to block a sensor, the voltage will decrease, and the sensor display will turn blue. When time has passed, and no changes have occurred, the sensor fades to gray, indicating no recent activity. In addition, researchers added a chime to the applet so they could recognize when a sensor had fired without having continually watch the screen while they are positioning targets in the reserved areas. In order to examine the question of someone entering a reserved space, researchers defined a set of seven test cases to be observed. These cases are listed out in Table 3.2.

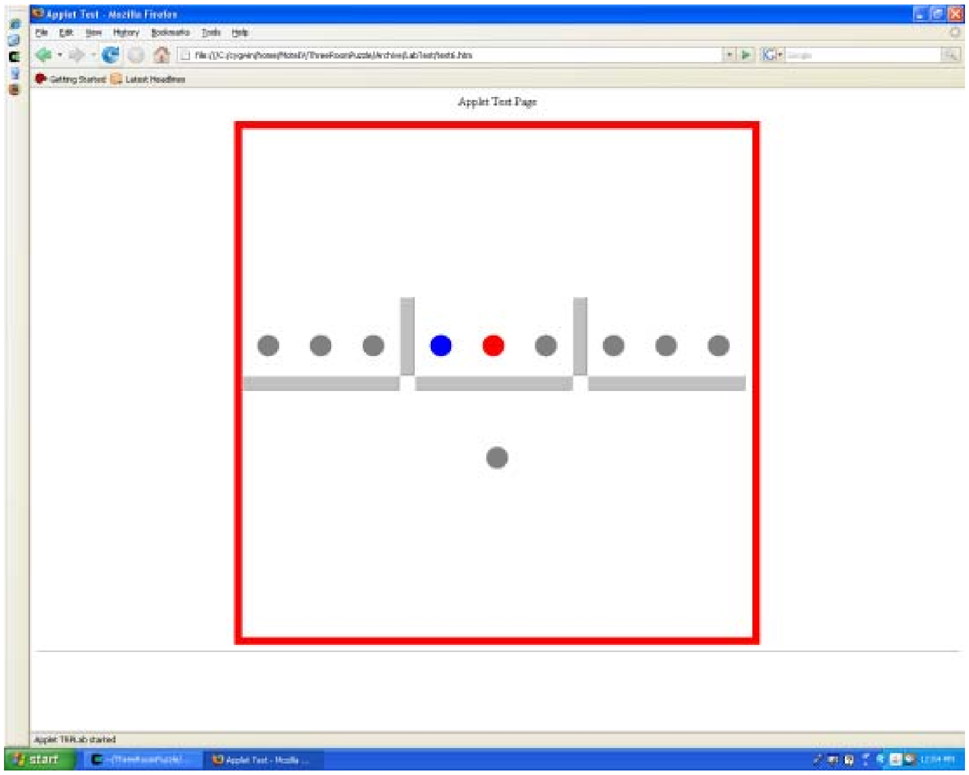


Figure 3.23 The TARGET applet.

For each test case, including the control, the sensors were brought online and a time was kept. After some time interval *t*,. another target would be placed into the test areas. Each test area would have zero, one, or two targets placed within the sensor detection field. The goal of the experiments was to show that the sensors did react and that the reaction could be measured.

Con denotes the control case in which nothing happens in any of the rooms. Then, case 1A considers what happens when a single target enters any of a series of empty rooms. The second two cases, 2a and 2b consider if there are two events. In one event, the reserved space is maintained, and the other there is a conflict. Case 3a and 3b illustrated the same situation, except there is an added event of a third person being in an adjacent reserved space. The final case, labeled Max, has four events, one target enters each reserved space, and then a fourth person violates a reserved area.

Table 3.2 Test cases for Three Room Puzzle

	Before			After		
Case #	Room 1	Room 2	Room 3	Room 1	Room 2	Room 3
Con	0	0	0	0	0	0
1A	0	0	0	1	0	0
2A	0	1	0	1	1	0
2B	0	1	0	2	0	0
3A	0	1	1	1	1	1
3B	0	1	1	2	1	0
Max	1	1	1	2	1	1

The results of these situations have been recorded using the applet and the WSN to store the data into files which have been graphed to look at the responses. In these examples, the first room contains motes 1001,1002, and 1003. The second room contains motes 1004, 1005, and 1006. The final room is 1007, 1008, and 1009. The tenth mote is a spare, and is not used in this configuration

Every 30 seconds, another event is added to the test case. For example in test case 1A, a single subject is added to the Room 2 just past the 30 second mark. The three different motes in that area respond with a spike in the data, followed by a new equilibrium. The results of this data can be seen in Figure 3.24and Figure 3.25. For test case 1A, none of the three sensors are triggering, because no target has been added. All three lines are flat. The red line, blue line, and black line represent three different sensors watching the left side, the back, and the right side of the area. In room 2 of the same experiments,, the lines are the same as in Figure 3.24. As a target enters the sensor range, the sensors spike, and then shift to a new equilibrium

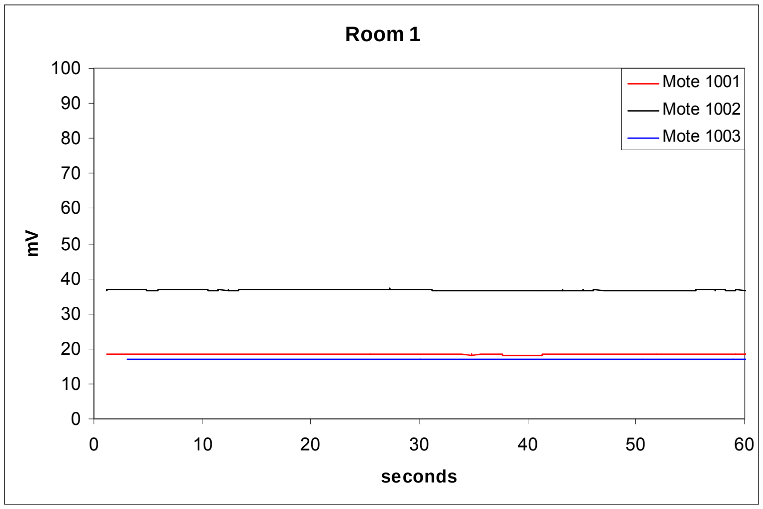


Figure 3.24 Room 1 in test case 1A.

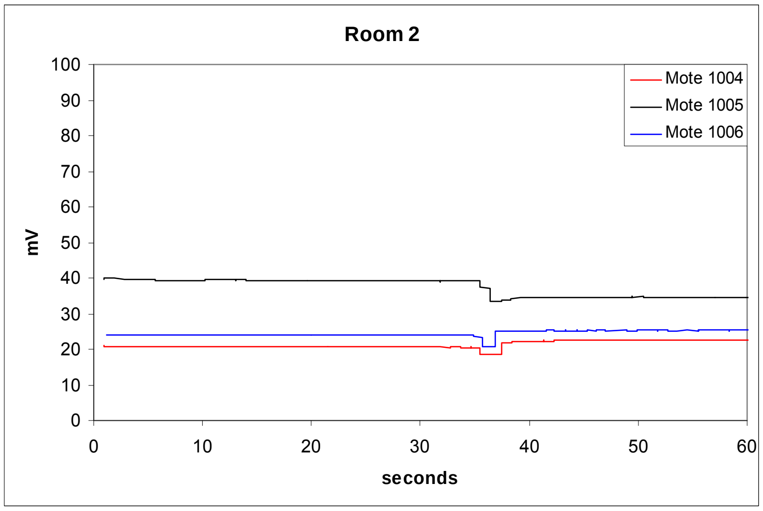


Figure 3.25 Room 2 in test case 1A.

While case 1A shows the results of a single event, a more interesting case with respect to time for the motes can be seen in the Max case. In Figure 3.26, Figure 3.27, and Figure 3.28, a series of four events happen in all three rooms. At 30 seconds, the first subject enters Room 1. At 60 seconds, the next subject enters Room 2. At the 90 second mark, the third target enters Room 3. At this point, each reserved area is now occupied. Then the fourth event occurs at 120 seconds, and Room 1 responds and has to settle to a new level.

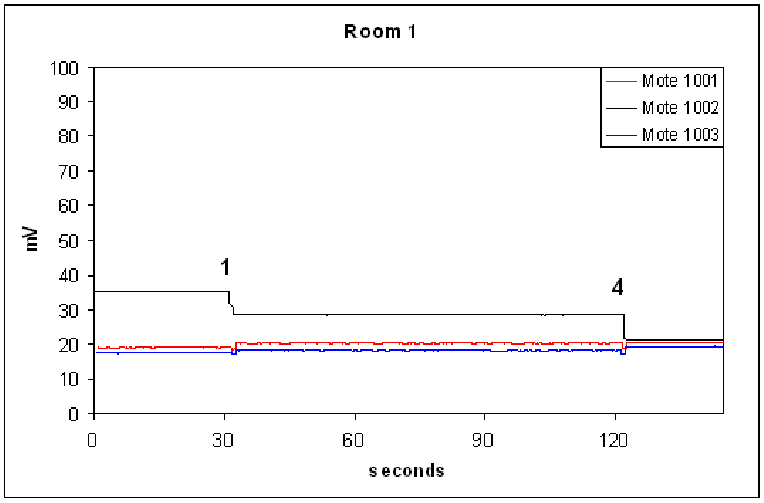


Figure 3.26 Events occur at 30 seconds and 120 seconds in Room 1

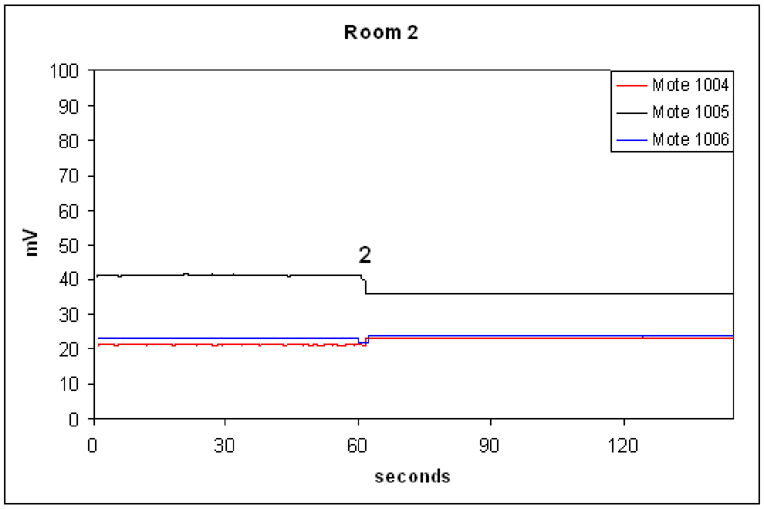


Figure 3.27 Event occurs at 60 seconds in Room 2

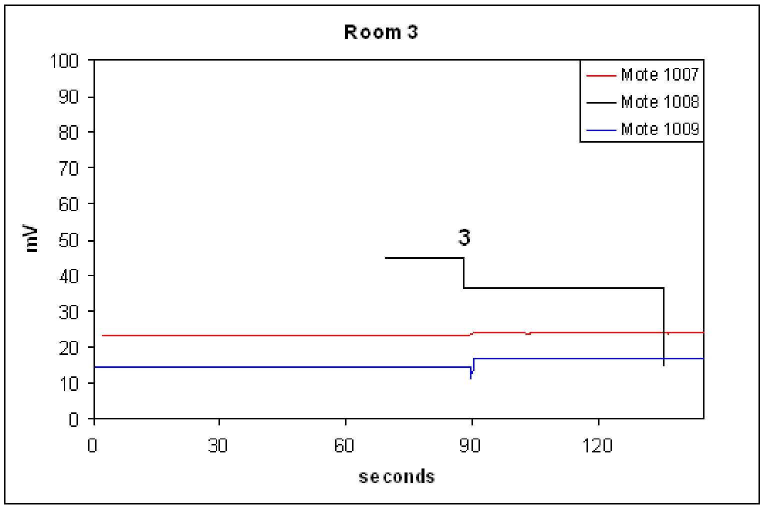


Figure 3.28 Event occurs at 90 seconds in Room 3

This set of cases demonstrate how the photocells respond to a new target entering their field of view and then level off after each event has passed.

3.2.3 Apartment Phase 1: COVERAGE

TARGET had only at most 4 events, 10 sensors, and ran for a maximum of 120 seconds. TARGET showed the reaction for a limited time for a limited area. We wanted an experiment closer to a human-sized environment. The Heracleia Laboratory has a apartment set up with cabinets and furniture. Motes were placed about the area for being able to watch a subject move on a path between the two rooms, as seen in Figure 3.29. The child motes containing the sensors were placed above a sample student work desk., while the gateway and the base station mote can be seen to the left on the table.

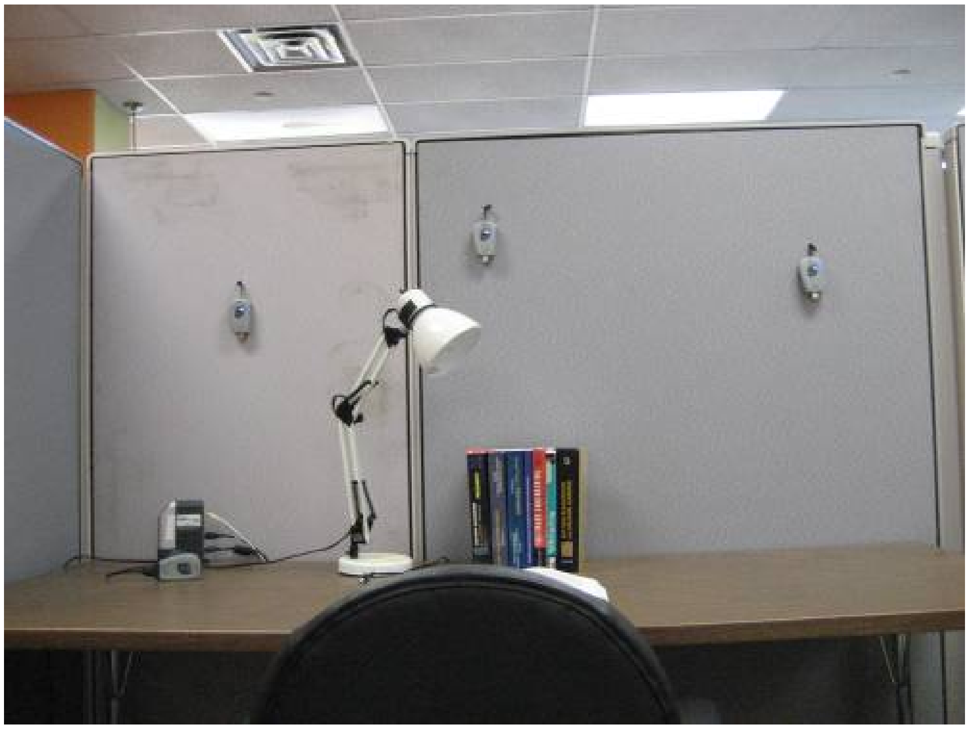


Figure 3.29 Child motes placed in the apartment.

Once the motes were placed in the apartment, their locations were marked down for reference, and a timer was started. The researchers then moved through the area to see if they could trigger events on the motes. The system was allowed to run for ten minutes as the researchers moved through the apartment as seen in Figure 3.30.

Also, the number of motes in use was doubled from 10 motes to 20 motes, and the applet shown in Figure 3.31 was reconfigured to match the new total. Each of these sensor displays corresponds to the sensor IDs in Figure 3.30. Now the system could handle 20 motes and run for a longer time period.

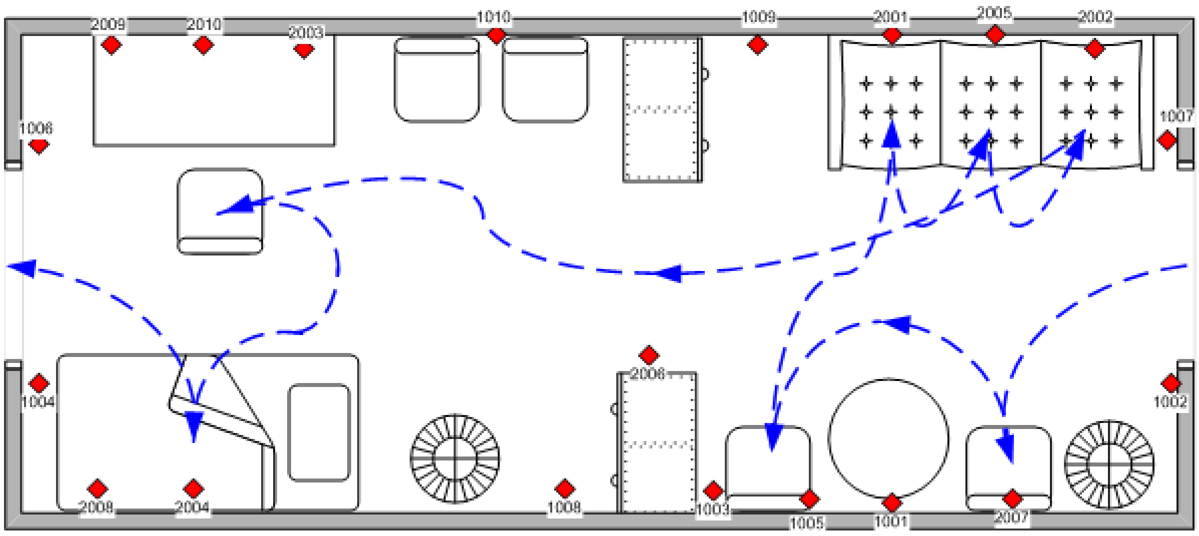


Figure 3.30 Moving through the Heracleia Apartment

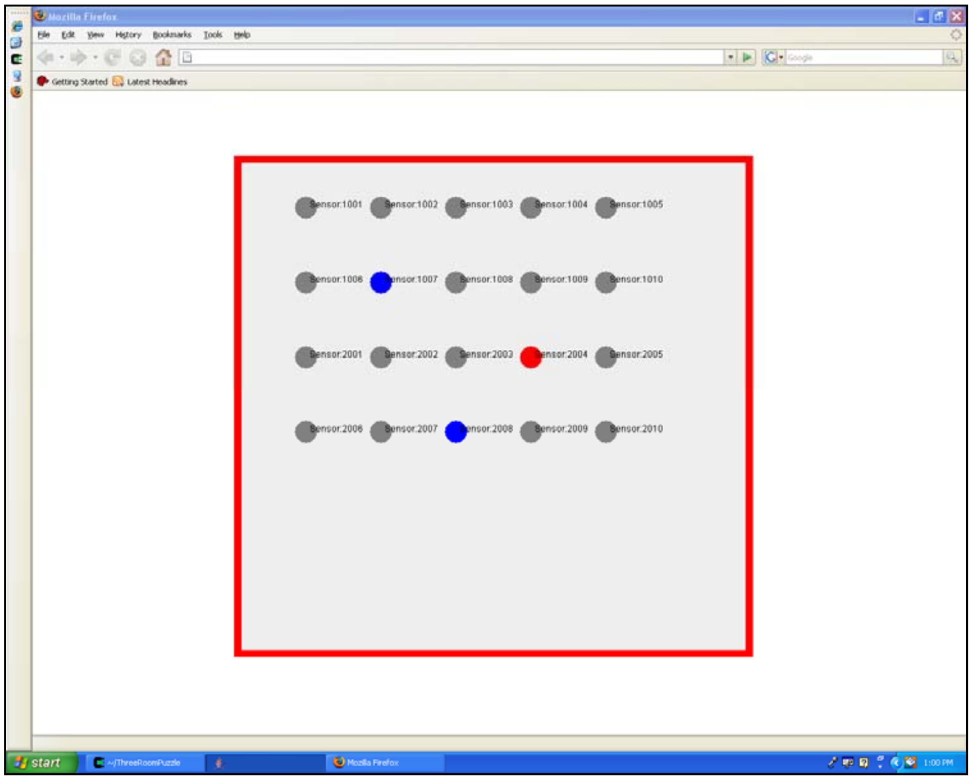


Figure 3.31 Applet configured for 20 motes.

After running the experiment, the photocell data from the various motes was graphed out to be examined. Most of the motes that were deployed would react as expected when someone moved by, as shown in Figure 3.32. The photocell reacted when the subjects moved through the area sitting on furniture near the motes.

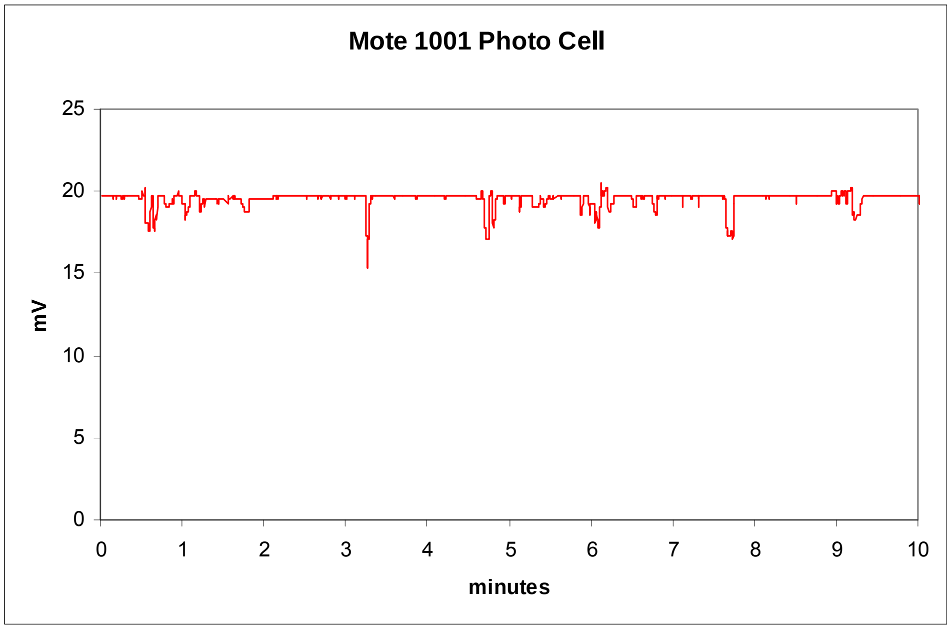


Figure 3.32 Mote 1001 reacts to a passerby.

However, not all the motes behaved as expected. Some motes that were in shadow by a tall cabinet did not appear to have enough light to sense a passerby. Also, motes that were parallel to the path of travel, such as mote 1007 in Figure 3.33, failed to signal an event or respond to the subject. None of the four motes watching the entrances registered events .

As a result of COVERAGE, we determined that the best reaction from the sensors occurred when the path of the subject passed perpendicularly to the sensor on the mote and to make certain they had enough light to register a voltage.

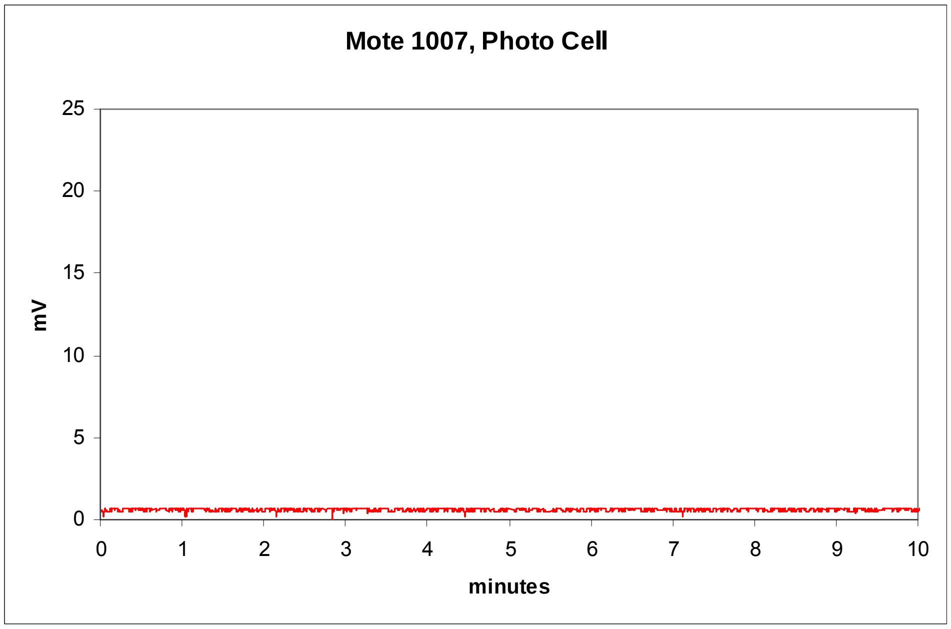


Figure 3.33 Mote 1007 does not respond

3.2.4 Apartment Phase 2: MOVEMENT

In addition to TARGET and COVERAGE, a pair of sample cases for motion were tested using the MOVEMENT sensor setup. The apartment was reconfigured with two rooms, a kitchen and a bedroom. These two rooms were connected by an adjoining hallway. Each of these three areas, the bedroom, the hallway, and the kitchen, were marked out with sensors as displayed in Figure 3.34. A subject would then move between areas of the apartment to trigger the sensors.

Three sensor motes were placed to observe the bed in the bedroom where the subject would start from. Three more sensor motes were placed by the cabinet, the shelf, and the refrigerator in the kitchen. Six sensor motes were then distributed to map out the hallway connecting the two rooms to an exit.

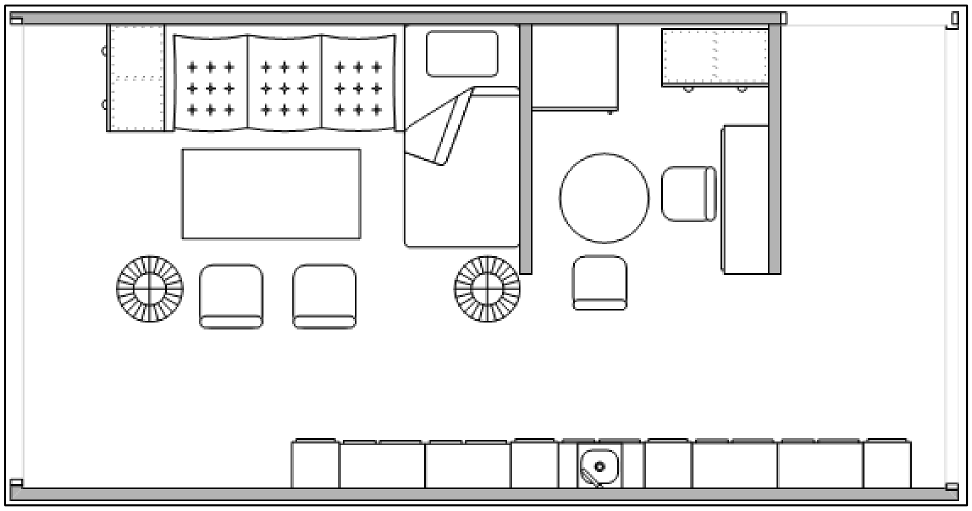


Figure 3.34 Second configuration of apartment

The applet was rewritten again, this time to display the layout of the new apartment. As before, the sensors would change color and chime to the changes in the environment, and a screen capture of this tool can be seen in Figure 3.35

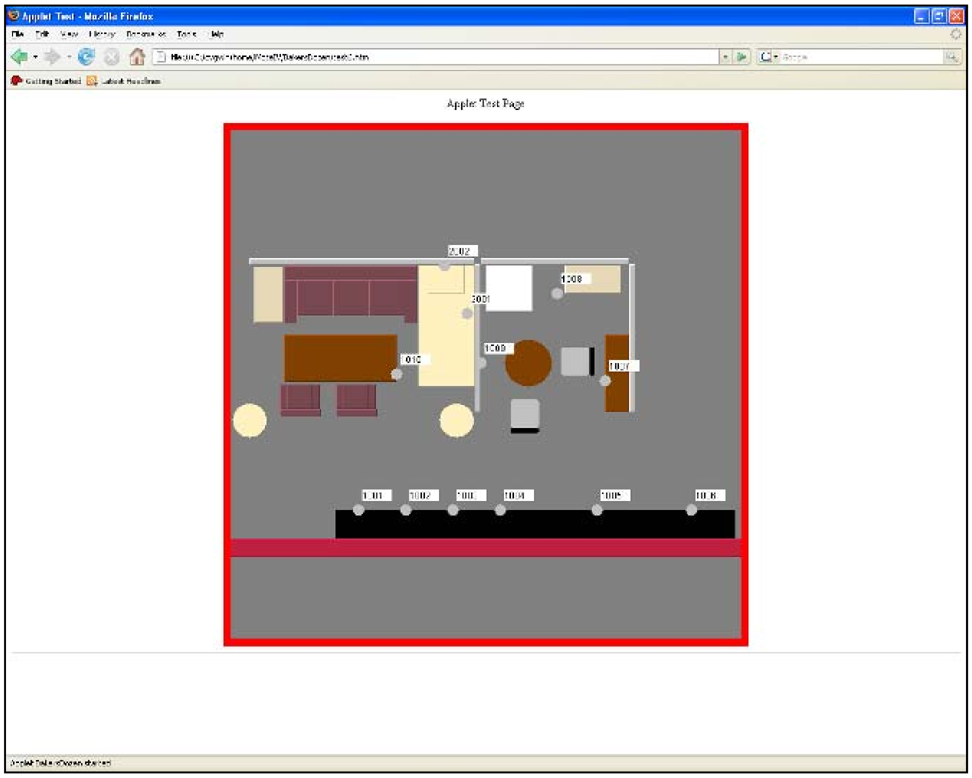


Figure 3.35 Applet for second apartment

3.2.4.1 *Bedroom to Kitchen*

The first experiment was of an individual getting up from the bed, leaving through the hallway, and arriving at the kitchen. As the subject passed the motes, the attached photocell sensor would record the change of light perpendicular to the path of motion. As the shadow of the subject falls on the light sensor, a lower voltage is recorded by the system. The data from these various sensors were then grouped by room and graphed by their color. **.** Blue sensors were for the kitchen, green sensors were for the bedroom, and red sensors were for the corridor.

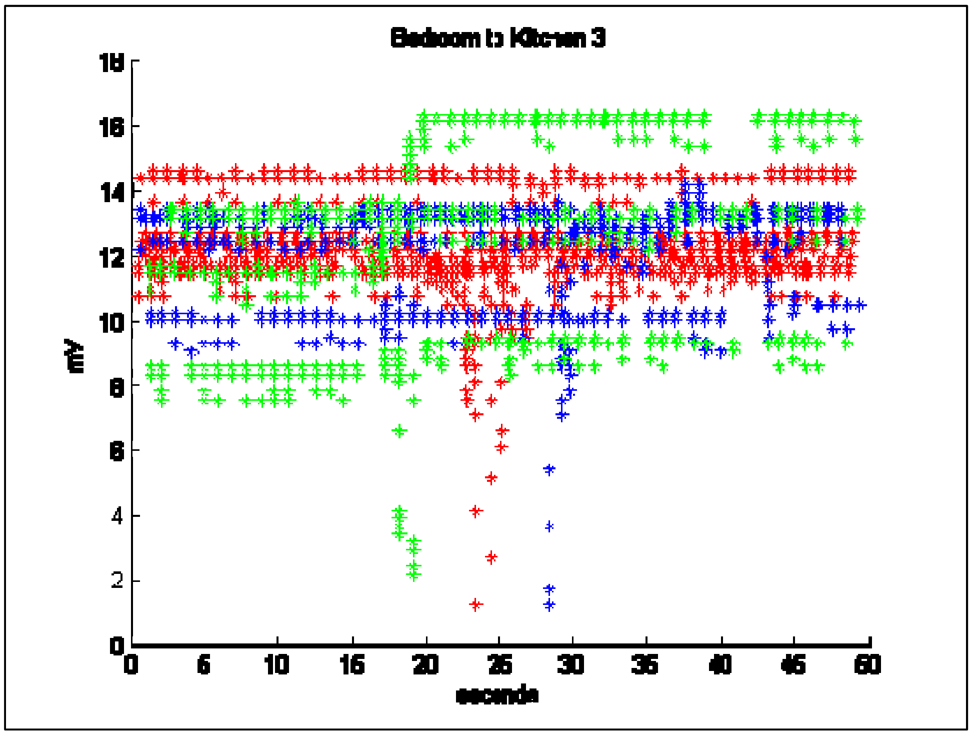


Figure 3.36 Sensors responding in three rooms

Figure 3.36 shows the motion of the subject in the bedroom, triggering the first set of sensors (which are in green). Next, the subject is moving out into the hallway and triggering the sensors watching the corridor that leads to the exit (which are in red). As the subject later enters the kitchen area, the third group of sensors register their presence (which are in blue).

3.2.4.2 *Bedroom to Exit*

A second scenario to be tested was what would occur when an individual left the bedroom and headed down the corridor towards the exit. In the apartment layout from Figure 3.34, the exit is on the lower right-hand side of the screen. As the subject walks down the corridor lined with sensor motes, they each record their progress as a drop in voltage. This sequence of voltage drops can be seen in Figure 3.37. Each sensor on each mote has been triggered in the order of their placement in a path that leads to the exit as the subject traversed the hallway. The mote response is graphed versus time, and the series of motes can be seen moving left to right through a series of motes as the subject moves towards the exit. The cross marking 1003 fires, then the square marker for 1004. Next is the diamond marker for 1005, and finally the star marker for 1006. Here, 1003, 1004, 1005, 1006 are the nodes deployed in the hall way.

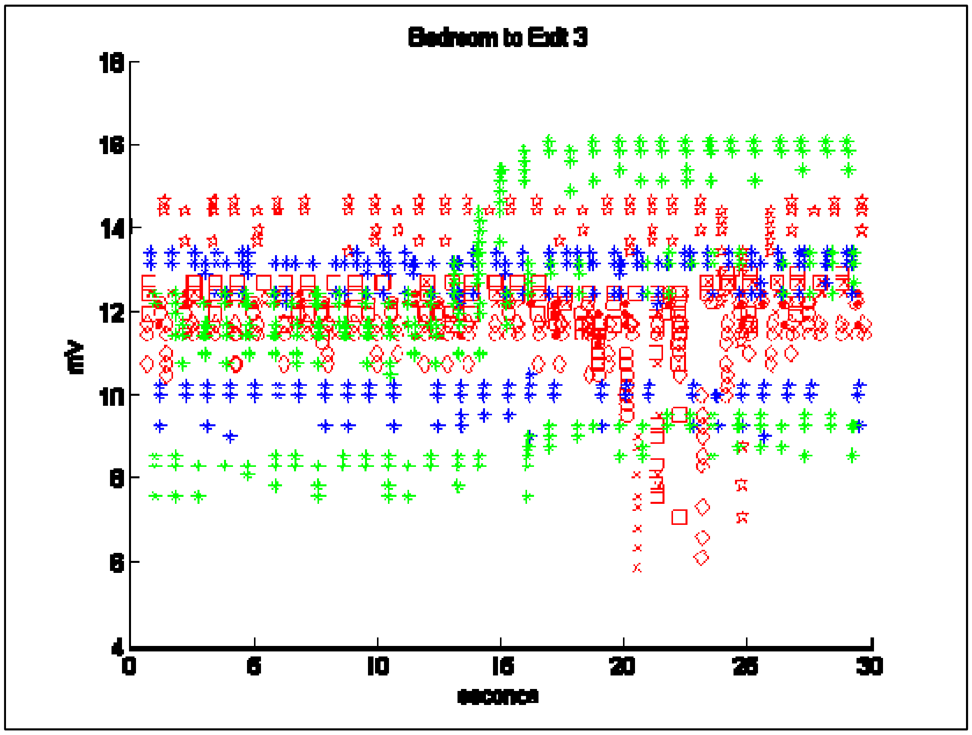


Figure 3.37 Sensors respond to motion down the corridor

3.2.5 Summary

As a result of these early experiments, the following can be observed about handling detection of events in an assistive environment. First, with TARGET, it is shown that the value recorded by the sensor is not the key component for detection. Instead, it is the change delta that is important instead. As each target enters the scanning zone, a noticeable change is noted before the system finds a new equilibrium. Second, with COVERAGE, the operating behavior of the sensors can be noted. In an environment with furniture and obstacles, as well as the moving subject, only the motion perpendicular to the photocell could be observed. Combining these two results allows the tracking through an assistive environment with the MOVEMENT program.

3.3 Event Analysis

Now, the ability to get data from the environment and how to place and handle sensor nodes has been established, new problems and new technologies entered this research. The T-Mote Invent sensor motes were no longer being manufactured. Instead, the SunSPOT nodes were selected to be the new research platform. As a result, new events and new sensors had to be investigated. In addition, the amount of data being broadcast from the sensor nodes to the base station (sink) reached a point where so much data was being lost that no activity could be monitored. To that end, the *episodic event* was introduced into overcome the bottleneck and the experiments began.

3.3.1 SunSPOT

The SunSPOT sensors consist of a small computerized board operating with a 2.4 GHz radio center, as can be seen in Figure 3.38. Each of these motes has a sensor daughter board with a set of different sensors. Each SunSPOT can recognize the voltage of the battery, the luminance of a photocell, the current air temperature from a thermistor, and the force affecting each spot in the form of the acceleration measured along the X,Y, and Z axis.[34]

Figure 3.38 SunSPOT motes

3.3.2 *Episodic Events*

In order to observe human activity within the apartment, it is necessary to find the changes in the sensor readings of the environment. But to gather all the raw data from the sensors and deliver them to a single collection point will completely consume all available bandwidth and result in the system being non-functional. To get around this situation, *episodic events* are defined and calculated on the individual SunSPOT motes instead of being calculated at the basestation. Episodic events are identified when the difference between the average and the current reading is greater than some established threshold t_k.

$$\left(x_j - \frac{\sum_{i=1}^{n} x_i}{n}\right) > t_k.$$

If the difference between the current event x_j and the average of the last set of *n* data points x_i is greater than the threshold t_k for event *k*, then we say the sensor has detected an event and the SunSPOT should signal the base station. In addition to the change in the data stream, a delay is also recorded to prevent a single event from being registered multiple times in a single sequence.

3.3.2.1 *Episodic Event Thresholds and Delays*

As described above, events begin as *episodic events*, meaning that each of these events represent a single activity impacting upon the sensors of a single mote. As a result, the mounting and placement of each mote has a direct effect on the reading of the sensors. As can be seen in Table 3.3 Episodic Event Parameters, twelve episodic events based on the WSN are defined, and then discussed in more detail.

Path Event-When using the T-Mote Invent sensor motes, sensors were placed to detect the passage of an object when its shadow fell across the sensor. Sensor Motes had to be placed upright and their photocell window facing perpendicular to the path of travel.

Bump Event-Bump events are used in three ways. Firstly, these would be attached to a piece of furniture to find out when a human bumped into it. Second, the mote could be used to represent the handle of a faucet or other latch so that its movement would indicate a human presence. Finally, the sensor could be worn by the human to see if they are still in motion. Bump events work on the magnitude of the three accelerometers in the x, y, and z directions and do not care how they are mounted.

Table 3.3 Episodic Event Parameters

Event Type	Event Number	Threshold	Units	Delay (ms)
Path	1	60	lux	500
Bump	2	0.125	g	1000
Furniture (Bed)	3	0.150	g	0
Battery	4	3.5	V	0
New Path	5	137	lux	500
Drawer	6	0.064	g	500
Vertical Hinge	7	0.15	g	5000
Horizontal Hinge	8	1.117	g	5000
Chair	9	0.909	g	100
Fall	10	2.25 (0.5)	g	1000
Switch	11	On/Off	N/A	1500

Furniture (Bed) Event – furniture events were developed to find out if a person has gotten on or off of a mattress or cushion in the course of their activities. These sensor motes respond to the acceleration in the y and z directions. These events occur on sensor motes that are mounted vertically, usually in a small pouch so they can rotate about the x-axis.

Battery Event-The battery event was the first status message sent by the motes during experiments. According to documentation, SunSPOT sensor motes are considered to be low on power when their battery voltage dips below 3.5 volts. This event does not require any special mounting.

New Path Event-When moving from the T-Mote Invents to the SunSPOT motes, the configuration of the photocell changed drastically. SunSPOT motes lack the protective shield and covering lens that come with the T-Mote events. These events come from motes mounted perpendicularly to the path of detection, and may even have an additional shield added to cut down on external light.

Drawer Event-Drawer events come from motes that are mounted vertically on the front of a drawer in a chest or a desk. The construction of a drawer limits the motion of the sensor in the x and y directions. For drawer events, the Z direction has the greatest range of motion and is the reading of interest for the drawer event.

Vertical Hinge Event-A vertical hinge event detects the motion that is from a hanging doorway, such as the door to a refrigerator or cabinet. The sensor is mounted vertically, parallel to the hinge. Due to the constraints of the hinge, only the motion in the x and z directions are considered relevant.

Horizontal Hinge Event-A horizontal hinge events detects the motion from a flap or lid, such as the flap of a desk or the lid of a toilet. In these cases, the mote is mounted 90' from vertical, and the readings are taken from the changes in the x and z axis of a mote. An addition to this event is the effect of gravity, which increases the threshold on the mote significantly.

Chair Event-Unlike the bed and bump events, a kitchen chair without wheels has to itself be moved in order for a human to sit down. As a result, the vectors of interest are in the x, y, and z directions. Since all directions are used to find a magnitude, the positioning of the mote is not important.

Fall Event-Fall events come solely from a sensor mote worn at the waist. Ideally, the sensor should be upright and in a holder. Experiments were done with a cell-phone holster worn at the waist. Should the human fall, we assume the shock to the body would also impact the sensor being worn. All three directions are of interest in the fall event case. The threshold can be lowered to detect other sharp motions, as well as just that of falling.

Switch Event-The switch event is a binary event, which is used to simulate the use of an appliance, a piece of electronics, or other switch inside the apartment test bed. Instead of scanning for the human presence, the action of the human pressing a button attached to the sensor mote triggers the event.

3.4 Cyberphysical Apartment Tools

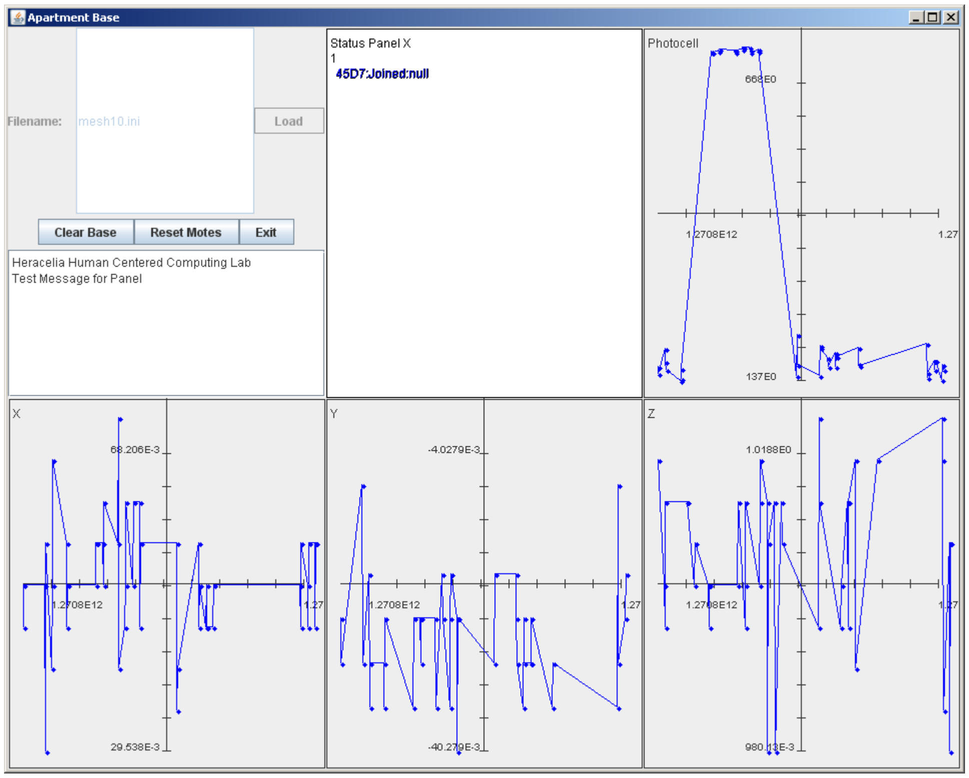
Over the course of the next phases of the apartment testbed, different tools using the new SunSPOT motes had to be developed and deployed in order to first find the data to generate the episodic events. Then tools had to be adapted to be able to record raw data, detect episodic events, and then display the episodic events. The first wave of these tools were developed for Phase 3 of the apartment, and later refined and updated for Phase 4 when new technology and devices were added.

3.4.1 Apartment Phase 3

For the third phase of the apartment, the SunSPOT motes were programmed to sample the raw data from humans interacting with furniture and other elements in their environment. Later, when the episodic events were defined, and a second tool was developed to record the events. [23]

3.4.1.1 *Raw Data Tool*

The first tool for the apartment is a raw data tool, which can be seen in Figure 3.39. The Raw Data tool allows a SunSPOT mote to connect and report their packets every 15 milliseconds back to the base stations, which is approximately 66.6 Hz, which should be a sample rate capable of capturing human activity. Each of the four main sensors, photocell, accelerometer x, accelerometer y, and accelerometer z can be seen and recorded while the researchers make experiments with the furniture and other elements in the apartment.



Status Panel X
1
45D7:Joined:null

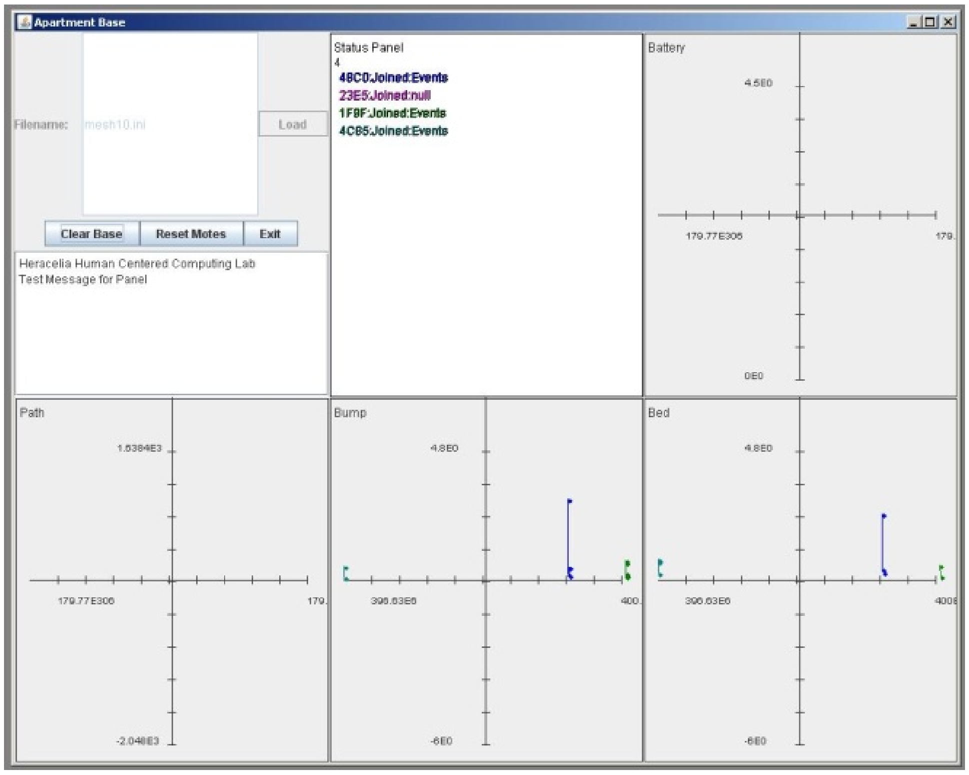
Heracelia Human Centered Computing Lab
Test Message for Panel

Figure 3.39 The Raw Data Tool

The results of these raw data collections can be saved to file, and then be analyzed later to determine what timing and levels of reaction are needed to detect an episodic event.

3.4.1.2 *Event Tool*

Each of the SunSPOTs broadcasts data packets to the ApartmentBase application, and the tool in turn stores the data to file for later examination. The data is also displayed on a graph for each type of data. One graph is generated for each accelerometer, the photocell, and the battery voltage. The later version of the tool has a graph for each key event.



```
Status Panel
4
 48C0:Joined:Events
 23E5:Joined:null
 1F8F:Joined:Events
 4CB5:Joined:Events
```

```
Heracelia Human Centered Computing Lab
Test Message for Panel
```

Figure 3.40 The Event Tool

These can be seen in Figure 3.40, where activity on the accelerometers are being displayed when a participant has interacted with one of the sensors in the WSN.

3.4.1.3 *EventMapper*

Another tool in our event-based system is the EventMapper application. Given the location and time each event occurs, the EventMapper can playback the responses of each mote in the apartment map as shown in Figure 3.41

The system reads in the various data files and sorts them into a sequence of events sorted by the time occurred. Once the event sequence is complete, the system can then respond by marking the map with the appropriate event in the right sequence. Each event can be assigned a different color, and the appropriate sensor display will respond with the correct color.

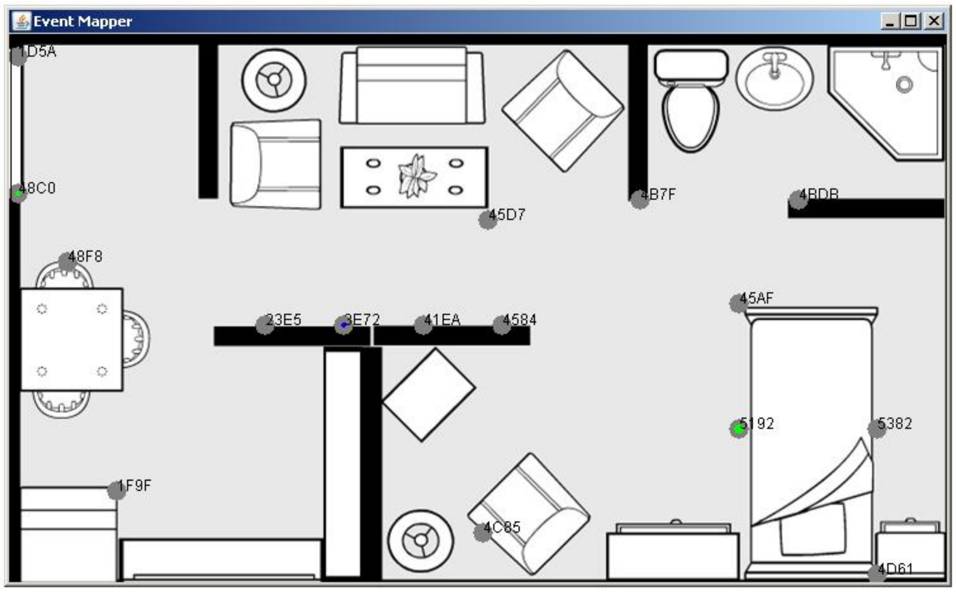


Figure 3.41 The Event Mapper Tool

The goal of the tool is to eventually be able to place the activity in a known location of the apartment by knowing the identity of the sensor being acted upon.

3.4.1.4 *Fall Detection Tool*

The events from the sensors placed in the apartment are static, and record the impacts or the changes in their environment. But another condition arose, a need to find out if someone had fallen in the apartment, which corresponds to the *events of interest.* The Fall Sensor is a wearable sensor, but has a similar behavior as the bump and bed events.

Emergency call switches that can be worn by an individual have been available for some time, but these are emergency units that have to be activated by the subject. But other shocks can be found from a wearable sensor. Assuming that the force upon the sensor can indicate the force upon the person wearing it, then an event should be able to be recorded that indicates if a shock has occurred. To do this, we applied the idea that a sensor that falls while attached to a human, then we can say the shock has occurred.

As with other events, this time the magnitude of the three accelerations is taken to determine if a fall has taken place. To find out what thresholds would be suitable, a sensor was dropped repeatedly from a height of three feet to the floor as described in section 4.4, on the idea that the sensor would be worn at the waist. Many smartphones and other devices contain accelerometers, and are carried on the hip of a subject. Perhaps such a device could be applied in a future experiment. The Fall Sensor actually has three different threshold levels as shown in Table 3.4.

Table 3.4 Fall Thresholds

Level	**Threshold Values (g)**
No shock	Under 0.75
Mild shock	Under 1.45
Medium Shock	Under 2.25
Fall	2.25 and above

The goal of the fall event is twofold. The first objective is to be able to record an event representing the force on the subject during the course of an experiment. This sort of event can be combined with the other sensor events to determine the behavior of the subject. Falls and shocks might not be reported unless it was a dire emergency, but could be a sign of dizziness or disorientation in the subject. A further application is to associate a severe physical shock reading with the motion in an assistive living apartment. If a shock occurs on the subject, and the bed, and in one room and no further events take place in the environment, an automatic help call could be placed to a caregiver.

When each of these levels is violated, a different magnitude is recorded on the Fall Event tool shown in Figure 3.42

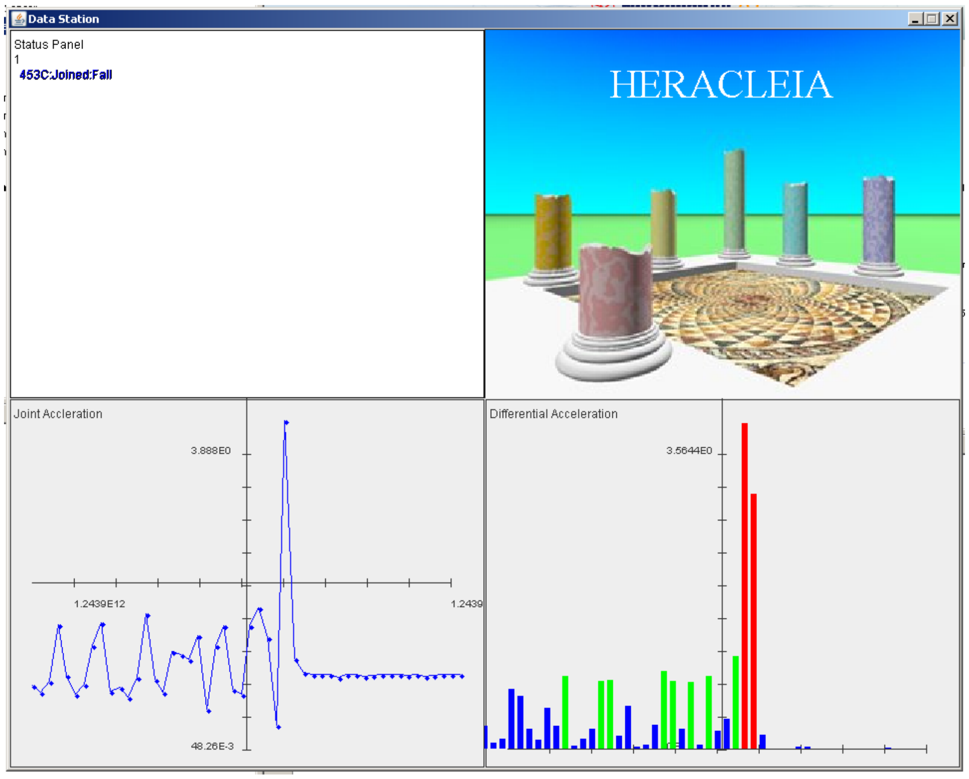


Figure 3.42 The Fall Detection Tool

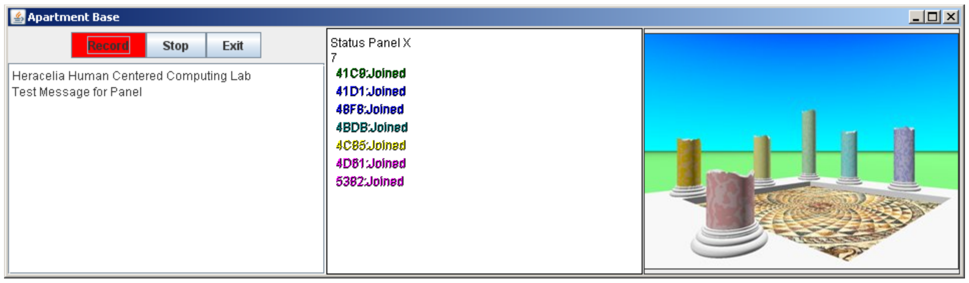
Once the worn sensor has joined the wireless sensor network, as seen in the upper left hand panel of Figure 3.42, the acceleration can be broadcast to the base station. The magnitude of acceleration is recorded on the lower left hand panel. The change in the acceleration, and therefore the force of the shock on the sensor, can be seen in the lower right hand panel. Ordinary movement and walking tend to show up in blue and green. Events of interest occur when the monitor moves into the red zone.

3.4.2 Apartment Phase 4

As time progressed, the apartment was updated again, and the tools and equipment had to be updated. Additional sensors, based on the Phidgets, and a FSA pressure pad were included into the apartment setup. A verification camera was put into place to record the proceedings, and the tools were updated to handle the new experiments as described below.

3.4.2.1 *Modal Event Tool*

In phase 3 of the apartment, every sensor would report all possible episodic events detected upon itself and its surroundings. In the beginning, this was acceptable, but as the system expanded to include more types of events and more pieces of furniture, this proved to be impractical. Instead, every SunSPOT mote had the ability to be configured to detect one particular type of episodic event.



```
Heracelia Human Centered Computing Lab
Test Message for Panel
```

```
Status Panel X
7
 41C9:Joined
 41D1:Joined
 48F8:Joined
 4BDB:Joined
 4C85:Joined
 4D81:Joined
 5382:Joined
```

Figure 3.43 The Modal Event Tool

For phase 4, the tool was upgraded to have a start and stop feature for recording of episodic events as seen in Figure 3.43. Now, the base station could be halted while the researchers and participants setup the apartment for the various phases of the experiment.

In order to facilitate the experiments in the apartment test bed, it was necessary to make the motes quickly configurable and resettable without erasing their current configurations. To do so, the switches on the motes were used to program extra features. The first switch on the mote would allow the builder to select which of the episodic events the mote should broadcast. The user could cycle through the options until the appropriate episodic event code would be displayed in binary on the SunSPOT mote, the numbers for which are in Table 3.5

Table 3.5 Episodic Event Codes

Event Type	Event Code	Binary LED Label
Path	1	0001
Bump Event	2	0010
Furniture Event	3	0011
Battery Event	4	0010
New Path	5	0101
Drawer Event	6	0110
Vertical Hinge	7	0111
Horizontal Hinge	8	1000
Chair Event	9	1001
Fall	10	1010
Switch	11	1011

In addition, connections to the various motes could be lost or disrupted by the experiment, needs of the base station program, or through random occurrence. In these cases, each disconnected sensor mote would have to rejoin the WSN while the builders were physically working in the test bed. To that end, the second switch would reset the state machine on the current sensor mote to return to the join state, and the builder can watch the status of the mote change on its LED display.

3.4.2.2 *Extended Event Experiments*

For the next set of experiments, we decided to limit the sensors to three key areas of our assistive living apartment: The bedroom, the bathroom, and the TV-watching area next to the bed as shown in Figure 3.44.

The first family of sensors used in this system is the WSN of SunSPOT motes from the previous phase of the apartment. The second family of sensors is wired sensors utilizing infrared sensors, pressure pads, and motion detectors. These sensors are configured with a central analog-to-digital control board that delivers their readings down a USB cable to a host machine. [35] These systems behave in a binary way, indicated the presence of a human within their operational area.

For the bedroom, the main sensors are the bed pressure pad and the pressure pads on either side of the bed. In addition, the desk, night stand, and dresser have also been fitted with sunspot sensors to detect the motion of the drawers and the writing flap. The desk flap has been fitted with a horizontal hinge event sensor. The drawers on the desk, nightstand, and dresser have been fitted with sunspots for drawer events. The bed is set up with three pressure pads. The first is the FSA pressure pad for high-resolution pressure data, and then on either side of the bed are two simple pressure pads for when people get on and off the mattress.

For the TV Watching area, the lounge chair is tagged with a motion detector to watch the combined bedroom and TV area. The space in front of the chair is equipped with a pressure pad to detect someone getting in and out of the chair. In one corner, an Infrared sensor has been placed to watch a broom. Also, four sunspots are active in this area. Upon the broom, a bump sensor is placed to detect if it is in motion. A New Path event sensor has been placed in the lamp to see if it turns on and off, and two more are on the television. One runs a switch event, to simulate turning the TV on and off. The second is another bump event sensor used to detect if the TV has been moved, should the user physically re-position the TV to watch it from another angle, such as the chair vs. somewhere else in the room.

The next area of the apartment is the bathroom, where another set of sensors have been deployed to monitor human activity. A Phidget board is in place to run the various wired sensors in the room. An IR sensor is placed to watch the activity on the waste basket by the toilet. The toilet is fitted with a horizontal hinge event to the seat lid, in order to detect the motion of the seat lid moving up and down. The sink has a battery of sensors. A pair of IR sensors watch for motion in front of the sink, while a third IR range finder scans a shelf-space that stores a towel. The taps on the sink, which is simulated, are actually two sunspots fitted as bump event sensors to detect motion. When a subject "turns on" one of the faucet taps, it will register on the base station. In the corner, a shower curtain is simulated using a foam divider, also fitted with a sensor to that we know when a user has entered to simulate showering.

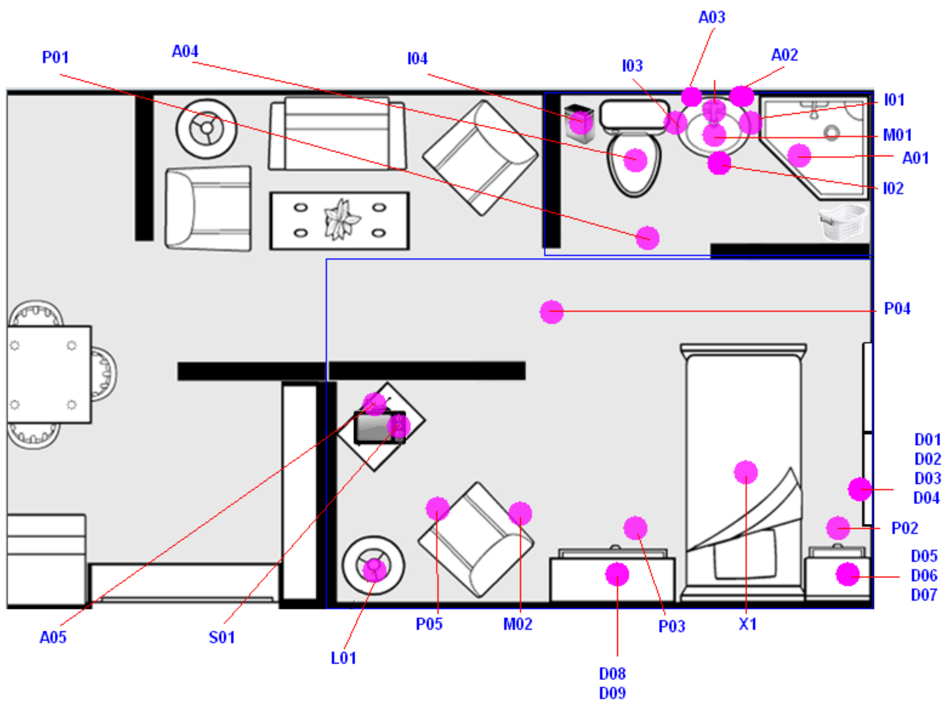


Figure 3.44 Apartment layout showing the sensor placement

Finally, there are the sensors carried by the test subject themselves. One sensor is configured to be a bump event, so that we can tell that the subject is always in motion. The second sensor is rigged to detected fall events, used for sensing the shock when a person should fall.

In addition to the camera was placed to record the events as they occurred within the apartment. The placement of the camera can be seen in Figure 3.45, where the view was concentrated upon the working area.

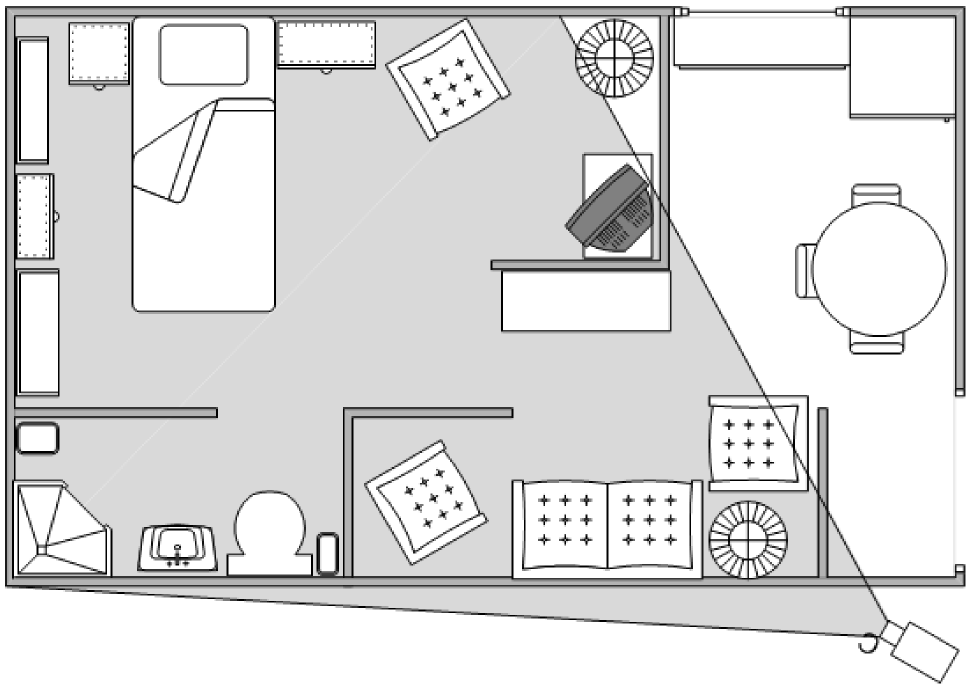


Figure 3.45 Placement of verification camera

3.4.2.3 *Human Activity Tool*

The Activity Analyzer in Figure 3.46 combines the ability to load the datasets, to load the descriptions of the activities, and then run the procedure to generate the listing of activities from the desired data set.

An expansion of the tool combines the ability to map a layout of the apartment, and to position each sensor upon a loadable floor plan. Additionally, each sensor can now be placed upon the floor plan and its properties and position can be adjusted using the mouse.

The tool is still under development at this time, but the end goal is to have a friendly user interface to allow the sensors to be deployed to the map via mouse and to be able to store and reload the configuration.

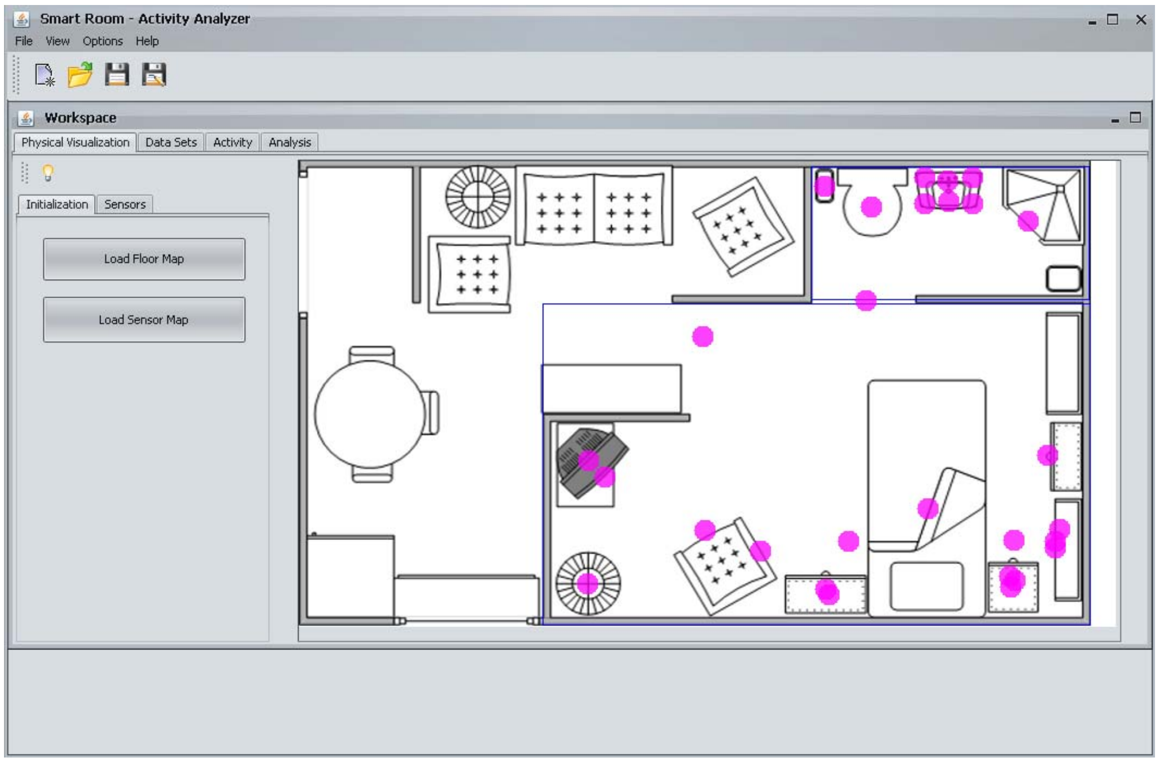


Figure 3.46 Human Activity Tool

3.4.3 Summary

These various phases of the apartment have shown an evolution of the cyberphysical tools that support the study of human behavior in the environment. As equipment, furniture, and appliances are added, the tools progress as well as the nature of the experiments. But it is also necessary to look at the role of who will use the data, and how it fits within the environment. For that, next we look at the Smart Drawer project and how it demonstrates the role of the human in such an assistive environment.

3.5 <u>The Smart Drawer</u>

3.5.1 Description

Within the assistive living apartment, various additional devices and projects have been started that work with the event detection pieces. One of the key projects is the Smart Drawer project, which is a medicine intake tracking system. The idea is that the medicine bottle being removed from a drawer will become an event that can be recognized and tracked within a cyberphysical setup. From this seed of an idea, the Smart Drawer began to expand. First, the RFID reader was started with the system, and then a *drawer event* was combined to make the next phase.

3.5.2 RFID Usage

RFID in healthcare is currently being used at hospitals for verifying the identity of patients. RFID technology is currently being used in hospitals for everything from tracking patients and infants, to making sure that the correct medication reaches the correct patient [36]. In order to streamline their auditing systems and for cutting down on costs, as well as being able to track the patient, hospitals are turning to RFID technology Using the RFID tags, the staff can verify the medication is received by the correct patient. The new tracking system also increases the accuracy of their billing over the older pen and paper system [37]. At hospitals, so many pills are issued per year that using RFID tags is expensive, but for critical items like IV bags, the RFID is extremely useful [38] . But RFID tags are now started being added to the medicine bottles in the supply chain by the pharmaceutical companies themselves [39]. Pharmaceutical companies have to take care, because they have to protect the safety of their products and their contracts by keeping track of their inventory. Companies like IBM have been deploying RFID tags with medication to prevent counterfeiting of medication in the drug supply [40], In order to block counterfeiting, major distributors of drugs are already putting RFID tags onto their bulk packaging [41][42]. The supply-chain for medications is already being tracked using RFID on the large scale. As the technology has been refined over the last decade, RFID tags start entering more and more into the supply chains of goods and services, especially anything manufactured including medication.

3.5.3 Implementation

The first version of the SmartDrawer was implemented using a and a series of experiments using ISO-14443 and ISO-15693 RFID tags with a TI 160 RFID Reader. The ISO-15693 tags were found to be able to support up to 8 tags at once and therefore were chosen for the first version of the SmartDrawer shown in Figure 3.47. [43][44]

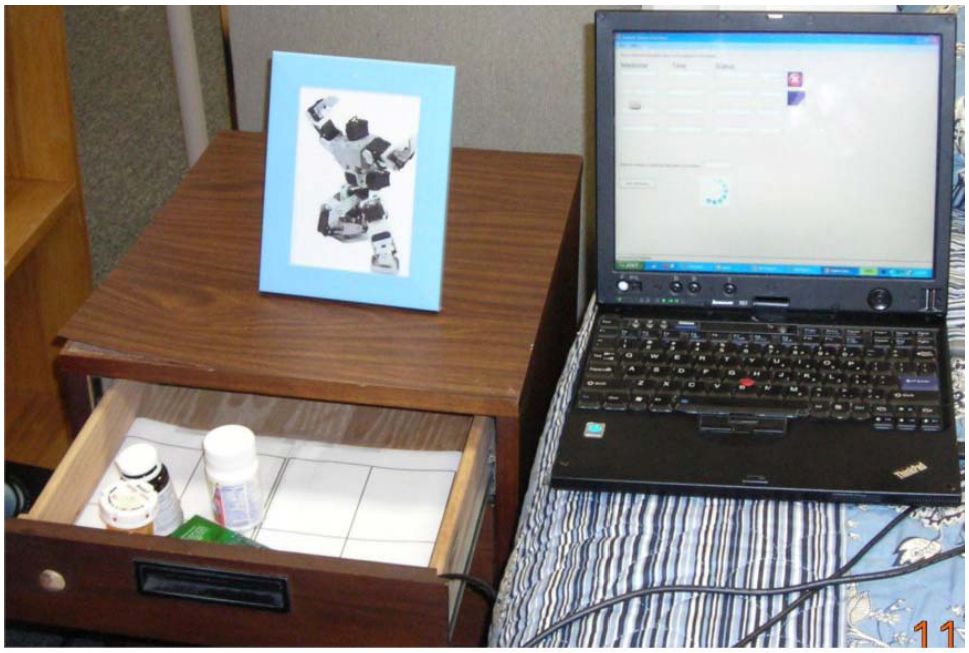


Figure 3.47 A Smart Drawer setup

From here, the concept was expanded to indicate that each removal of a bottle from the drawer would in turn be its own cyberphysical event, similar to an episodic event. With each time the drawer is opened and a bottle is removed, a time stamp can be taken from the activity and recorded for further use. For example, given the times of the day that medicine is taken, then a human or a machine would be able to find out if these were within the specification of the prescriber.

3.5.4 User Roles

For event detection, the SmartDrawer project defines certain key roles for a user in their interaction with the cyberphysical environment. One of the uses of event detection is the application of the system over a period of time. To that end, three user roles have been defined for use with the SmartDrawer Project.

3.5.4.1 *The Patient*

The first role is the person who is using the system for reminders and the intake of medication, the *patient.* The goal of a cyberphysical system in this case is to detect the events over the course of the day, whether they be the removal of medicine from the Smart Drawer or if they have a reminder system with a schedule that prompts them to take their medicine. Other options would include the ability to inform the patient if the right bottle has been selected. Unlike a medicine dispenser [45] which requires a device to be preloaded, the goal would be to include a system that could operate off of a combination of the physician's prescription and the inventory tag arriving from the manufacturer. The time range involved for this role indicates an hourly granularity in which the system would respond.

3.5.4.2 *The Caregiver*

The second role is of a nurse or family member who is keeping track of the medication to make sure the patient is following the prescription, and has to know how the patient is doing on a daily basis. This role is defined as the Caregiver, who is responsible for checking up to see if the patient has had problems during the day and to make sure that the medicine intake was according to the prescription. In order to take a pill three times a day, the log of the medication must be recorded over the course of the day. Then, the Caregiver can keep this for medical records or the daily medicine intake can be compared to the prescription to see if a dose has been missed.

3.5.4.3 *The Maintainer*

The third role is the Maintainer, who is someone who wishes to track the pattern of medication intake of a period of time, at least a week or more. The Maintainer would have access to the database, and be able to update and upgrade the system as necessary. In addition, the Maintainer would be able to access a log of the medication intake over time and compare this to other detected events recorded in the database to find correlations between human activity and medicine intake.

3.5.5 Summary

The Smart Drawer project illustrates the concept that any cyberphysical system would have to have defined roles for the users of the system. The two main defining aspects of these roles are the security, who should have access to the data, and temporal, what is the time frame in which the data is to be collected or acted upon.

CHAPTER 4

DISCUSSION

For purposes of discussion, several additional topics have been added to this work. These topics range from topic to topic, but all refer back to the wireless sensor network. First is the question of network failure, and how to come up with a model to measure reliability. Next, there is the question of the validation of episodic events. A recent topic in both current events and in networks is the topic of software self repair, and how it could apply to a wireless sensor network. And finally, the role of the human in cyberphysical systems will be revisited.

4.1 Network Reliability

When a WSN has been deployed in an environment, it will encounter certain risks that can impact the reliability of the system. Risks include damage, bugs in programming, blind spots in the sensor network, interference from electromagnetic sources, sabotage, or power drain. [46]. The problem is how to measure the quality of service of wireless sensor networks.

WSNs quality includes measures of reliability, coverage and connectivity [47]. In recent years, WSNs have lowered in price, increased in computational capacity, and their motes have reduced in size. [48]. To maintain quality of service (**QoS**) to ensure a properly functioning network, radio links and motes are always prone to failure, either through power loss, obstacles, or other conditions 49. The failures in WSN can be studied with a stochastic approach if assumptions are made and analyzed by using a theoretical approach and simulations [50]. WSNs need to be able to handle their tasks while consuming their battery efficiently. As sensor motes register changes in their environment and report via radio, they consume their own battery life. [51]. When the voltage drops below operating capacity, the links in the WSN can be broken. The use of probabilistic mathematical models is one method to analyze the reliability of sensor networks. Estimations can be made about the properties of the WSN by using such a model. The location of the nodes in a WSN has a geometry that can be to test the coverage and probability of failure in the network [52]. To do this for the question of reliability, a function is defined for the WSN, by assuming two basic structures in the network. Then, the function is applied to a series of test cases.

4.1.1 The Problem

A WSN is modeled as a graph composed of a number of sensor nodes and a computer server at the root. Assuming that when a sensor mote detects an event, it broadcasts to the base station though other nodes. These nodes in turn are physically closer to the base station. .

Assuming:

- The base station never fails;
- All the sensor nodes are homogeneous;
- The routing algorithm is single path routing [53], which means a node broadcasts only to a single neighbor node.

Assuming the base station never fails allows the function to concentrate on the sensor motes. Allowing the sensor motes to be homogeneous allows the consideration of sensor lifetimes with independent and identical distributions (i.d.d.). By limiting the message forwarding to a single neighbor node rules out loops and other cases in the WSN graph.

The model of the lifetime of a sensor node is an exponential distribution, i.e., X ~ E(λ) where X is the lifetime of a sensor node, λ is the parameter for this exponential distribution of the sensor lifetime, and the probability density function *p(x)* of E(λ) is as follows:

$$p(x) = \begin{cases} \lambda e^{-\lambda x}, x \geq 0 \\ 0, x < 0 \end{cases},$$

and its distribution function (*i.e.* cumulative distribution function) is

$$P[X \leq x] = \int_0^x p(x)dx = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}.$$

The physical meaning of λ is the reciprocal of the mathematical expectation value of the lifetime of sensor nodes. Namely, the greater λ is, the shorter the sensor lifetime is expected.

The reliability function of the system represents the robustness of the WSN to a sensor mote failure. To measure WSN reliability, the reliability function *R(x)* is defined as follows: *reliability* is defined as the probability of the system's survival up to moment x, *i.e.*,

R(X) = P[there are no failure events over time [0,x]].

4.1.2 The Model

The approach taken to model the WSN is called the *Component-based Network Reliability*(**CNR**)

For the CNR, two types of components are defined: serial and parallel. The goal is to allow a WSN network to be decomposed into these two cases.

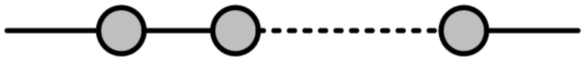


Figure 4.1 Sequential Nodes

The first case, as seen in Figure 4.1, shows the nodes are connected in sequence, a sequential connection where if one sensor fails then the entire system fails. The reliability of the network is represented by the probability of failure of a node over some *time horizon x*, assuming that each sensor fails independently and the lifetime of the sensors are identically distributed. According to the reliability function, the *sequential reliability function* is defined as follows,

$$\begin{aligned} &R_s(x) = P[\min(X_1, X_2, ..., X_n) > x] \\ &= P[X_1 > x, X_2 > x, ..., X_n > x] \\ &\overset{i.i.d.}{=} \prod_{i=1}^{n} P[X_i > x], \end{aligned}$$

Where n is the total number of sensor nodes in the system.

Now let us consider the case of *exponential distribution*. The following holds:

$$P[X_i > x] = 1 - P[X_i \le x] = e^{-\lambda x}.$$

From the above we find that

$$R_s(x) = e^{-n\lambda x}.$$

The second case is when nodes are connected in parallel, as can be seen in Figure 4.2.

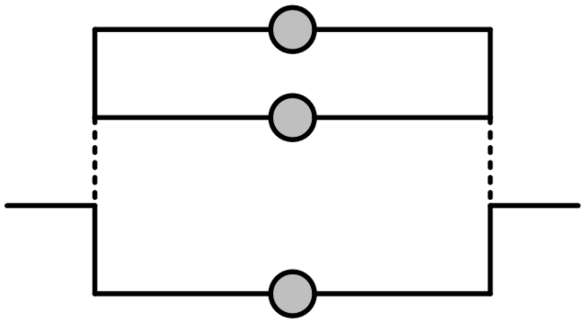


Figure 4.2 Nodes connected in parallel

In this case, the network will only fail when all the nodes of the component are broken. The *parallel reliability function* is defined as follows:

$$\begin{aligned} R_p(x) &= P[\max(X_1, X_2, ..., X_n) > x] \\ &= 1 - P[\max(X_1, X_2, ..., X_n) \leq x] \\ &= 1 - P[X_1 \leq x, X_2 \leq x, ..., X_n \leq x] \\ &\overset{i.i.d.}{=} 1 - \prod_{i=1}^{n} P[X_i \leq x] \\ &= 1 - (1 - e^{-\lambda x})^n . \end{aligned}$$

These two components represent the two extreme cases of the CNR, where either a single node will disable the network, or when all the nodes must fail to block a message packet.

These two formulas can be used as the upper and lower bounds for the reliability function for an arbitrary minimal number of sensors in the functional state of the system,

$$e^{-n\lambda x} \leq R(x) \leq= 1 - (1 - e^{-\lambda x})^n .$$

Combining the two basic components described in and results in a new network shown in Figure 4.3. Two nodes , X11 and X12, are connected in parallel while nodes X2 and X3 continue in sequence. In this case the failure of one of the nodes does not produce failure of the system.

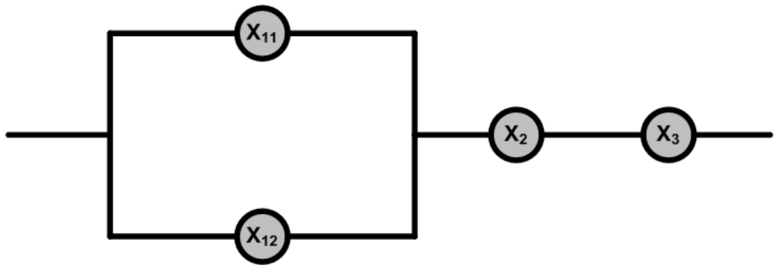


Figure 4.3 Combination of parallel and sequential components

In this combination, the reliability function can be calculated as follows,

$$\begin{aligned} R_c(x) &= P[\min(\max(X_{11}, X_{12}), X_2, X_3)] > x] \\ &= P[\max(X_{11}, X_{12}) > x, X_2 > x, X_3 > x] \\ &= (1 - P[\max(X_{11}, X_{12}) \le x])P[X_2 > x]P[X_3 > x] \\ &= (1 - (1 - e^{-\lambda x})^2)e^{-2\lambda x}. \end{aligned}$$

4.1.3 Examples

A popular WSN structure in practice is a tree, and to study reliability, they can be fitted with redundant nodes and the reliability function applied.

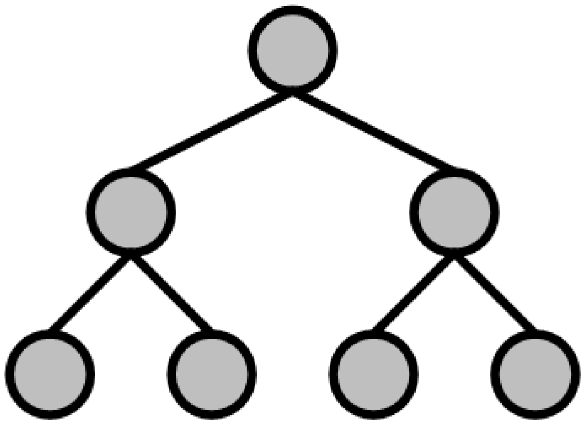


Figure 4.4 Tree with no redundancy

Figure 4.4 is a three-layer tree, and each node is singular, and in this structure, every node is equally important. Assuming every spot in the monitored area is equally important, the tree behaves a a sequential case with regard to the reliability analysis. The reliability function for the seven-node case is:

$$R_7(x) = e^{-7\lambda x}$$

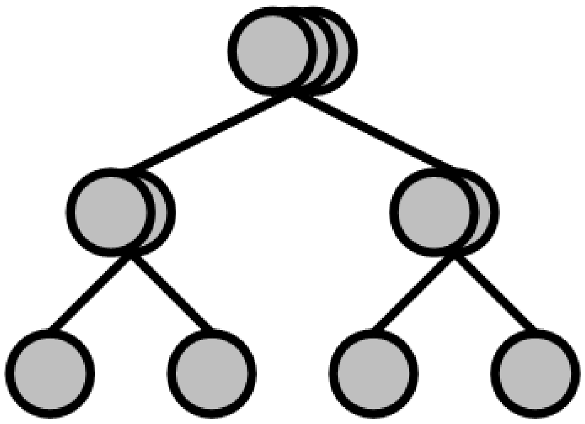


Figure 4.5 Stepped redundant case

Figure 4.5 gives another three-layer tree, where a second-layer node has a backup node, and the root node has three redundant nodes. Each group of duplicated nodes logically equal to a parallel case, and the logical connection between each cluster is sequential. So, the eleven-node reliability function is:

$$R_{11}(x) = e^{-4\lambda x}(1-(1-e^{-\lambda x})^2)^2(1-(1-e^{-\lambda x})^3).$$

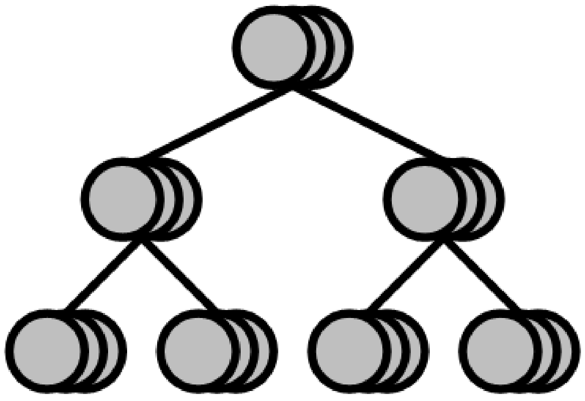


Figure 4.6 Fully redundant case

In the tree structure of Figure 4.6, every node has three duplicate nodes. So the twenty-one-node reliability function is:

$$R_{21} = (1-(1-e^{-\lambda x})^3)^7.$$

4.1.4 Assistive Testbed

In the assistive testbed experiments [26][22][7][54], an apartment has sensor nodes located in various rooms and covering areas where human monitoring needs to take place. The SunSPOT motes were programmed to monitor certain human activities, such as opening a door, getting onto the bed, and passing through a hallway. The empirical average lifetime for a SunSPOT mote battery set to detect episodic events is approximately 3 days. Assuming λ=1/3, simulations run using MATLAB produce the reliability analysis Figure 4.7. R7 is the seven-node case, R11 is the eleven-node case, R21 is the twenty-one-node case, and R14 has the same structure as R21 except every cluster just has two duplicated nodes.

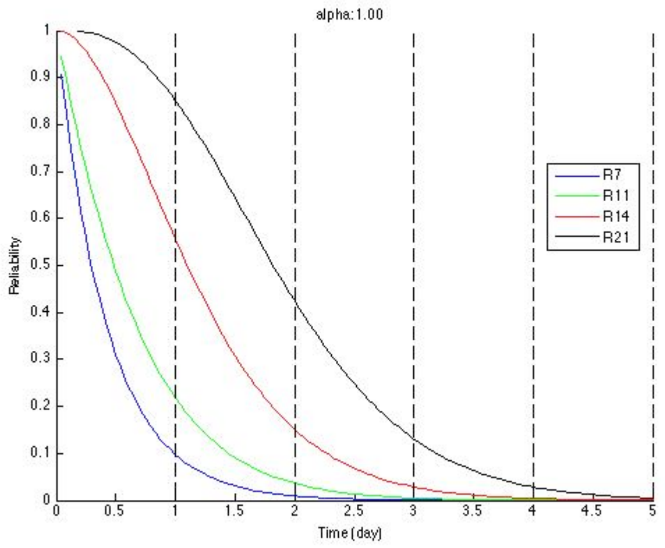


Figure 4.7 Reliability Function of Tree Structures

Additional CNR system modules could be designed to perform preliminary user-need modeling before proceeding with the reliability design. Expansions include the analysis of WSN availability and serviceability [55] in an assistive environment and the tradeoffs between physical redundancy and cost as protection against service interruptions.

4.2 <u>Validation of Events</u>

One topic for discussion is the behavior of the sensors for validation purposes. The use of accelerometers and photocells to capture human activity is not a purely binary answer, though they can be considered as such. The system could be altered to use solid-state switches and magnetic latches to identify activity. And even the Phidgets with their floor pads and other sensors yield a binary answer. Either the pad is active, or it is not.

But in the case of the SunSPOT motes, an additional review of their behavior is in order. The episodic events can be broken down into four categories. Photocell events, accelerometers with a fixed range of motion, external sensors that react to the entire human body, or worn sensors that are with the human body. The battery event and switch sensor events are not considered, since they have a fixed behavior.

4.2.1 Photocell Events

The first set of events are the photocell events, which were originally designed to use the TMote invents from {Add Section Here} With the change over to the SunSPOT motes however, the Path Event would no longer suffice. The difference between the two technologies was physical, with the TMote Invents including a shade and a lens to keep extraneous light from impacting upon the photocell. When the same event was applied to the SunSPOTs, the result looked like Figure 4.8 , where far too many false positives were detected.

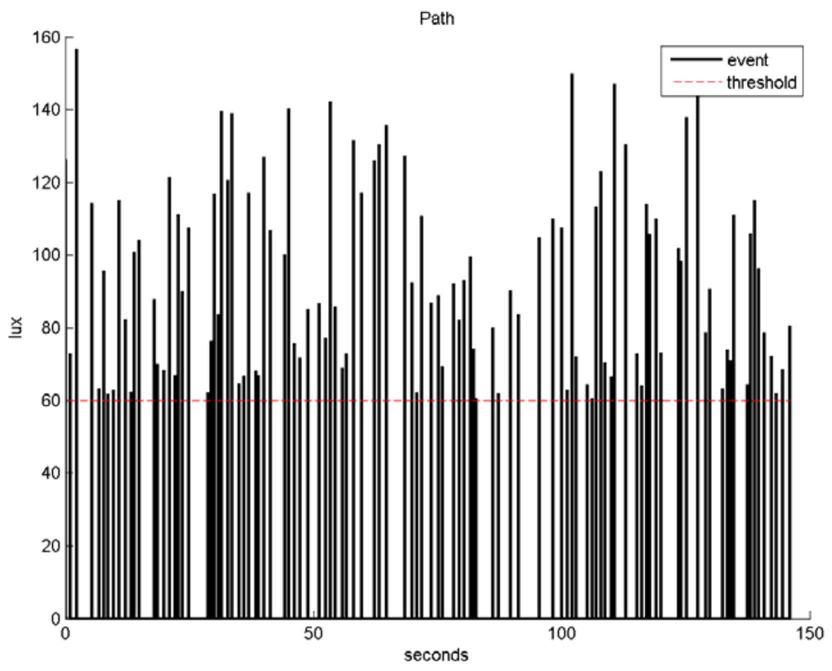


Figure 4.8 Path Event on SunSPOT

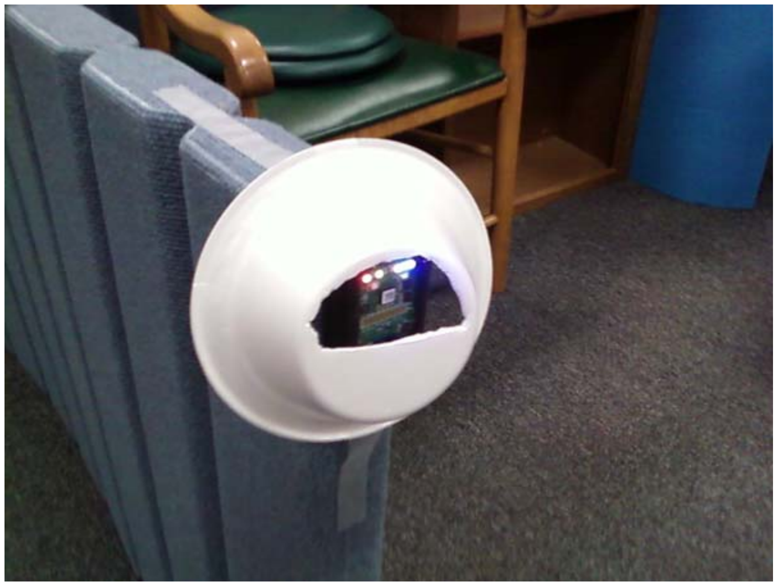


Figure 4.9 Adding Cover for Sunspot

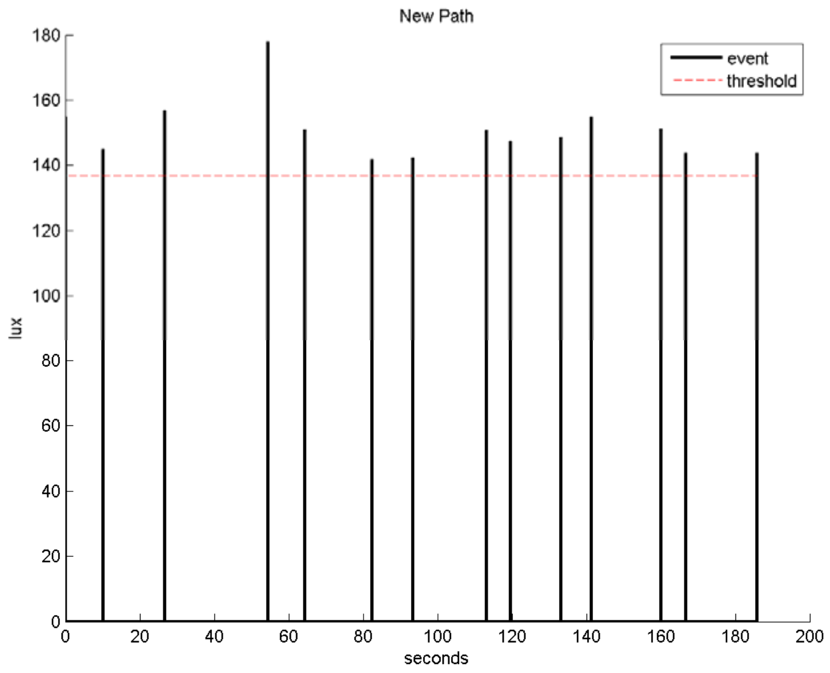


Figure 4.10 New Path Event with Cover

To counter the problem, a shade was added to the SunSPOT mote to cut out unwanted light (Figure 4.9). Still, the system did not behave predicablly, and the need for the additional cover makes the SunSPOT hard to deploy rapidly. The amount of false positives, however, decrease remarkably as seen in Figure 4.10.

4.2.2 Limited Range of Motion Events

Other events include the use of accelerometers, and one group of these events involve attaching the SunSPOT to piece of future with a moveable component, such as a drawer, a lid, or a door. These events are called, respectively, the drawer event, the horizontal hinge event, and the vertical hinge event.

Of the limited range of motion events, the drawer event is one of the most consistent, as can be seen in Figure 4.11. The limitation of the motion in a single direction seems to make the event more easily detectable.

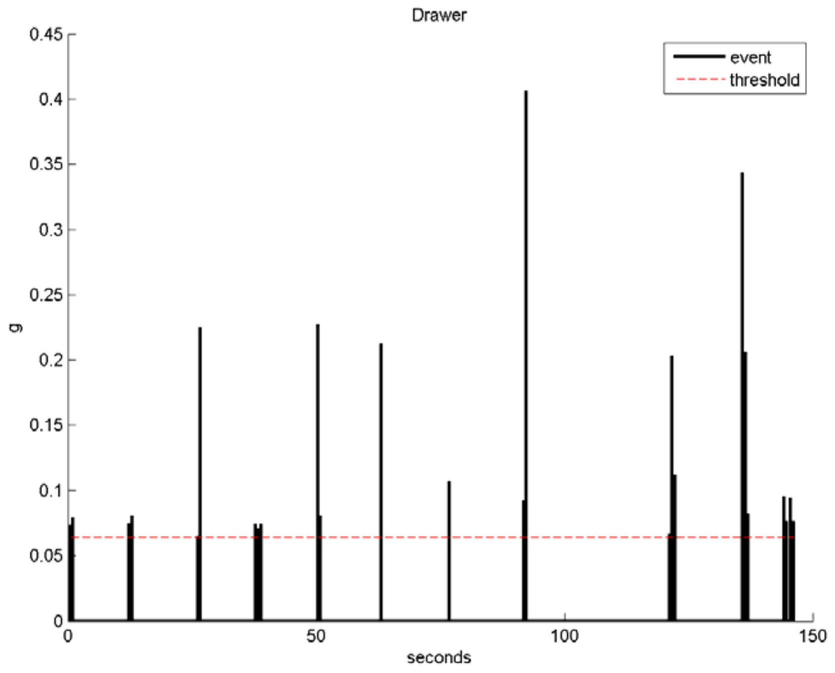


Figure 4.11 Drawer Event

Another event with a limited range of motion is the horizontal hinge event, which is most useful for items that are a hinged lid or flap. The accelerometer has to act against the pull of gravity, making the signature of the event large and clear. A series of the horizontal hinge events is shown in Figure 4.12 .

The vertical hinge event (Figure 4.13), or door event, did not seem as accurate as the first two, since it had a larger range of motion and no gravity. But the experiment was being done on the door of a refrigerator. The metal of the door blocked the radio signals until the base station was moved to be clear of the obstruction. As a result, the signals were only received about 70% of the time.

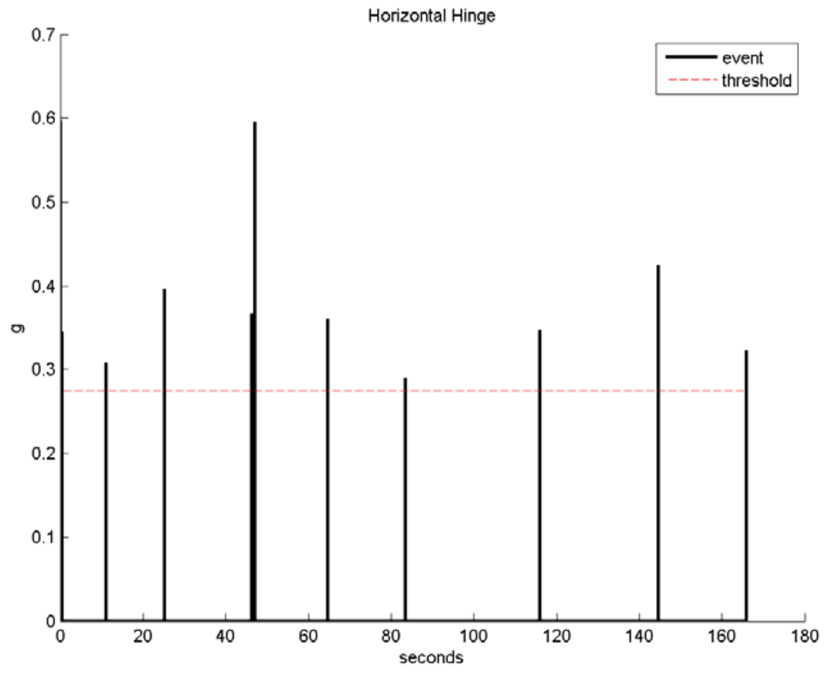


Figure 4.12 Horizontal Hinge Event

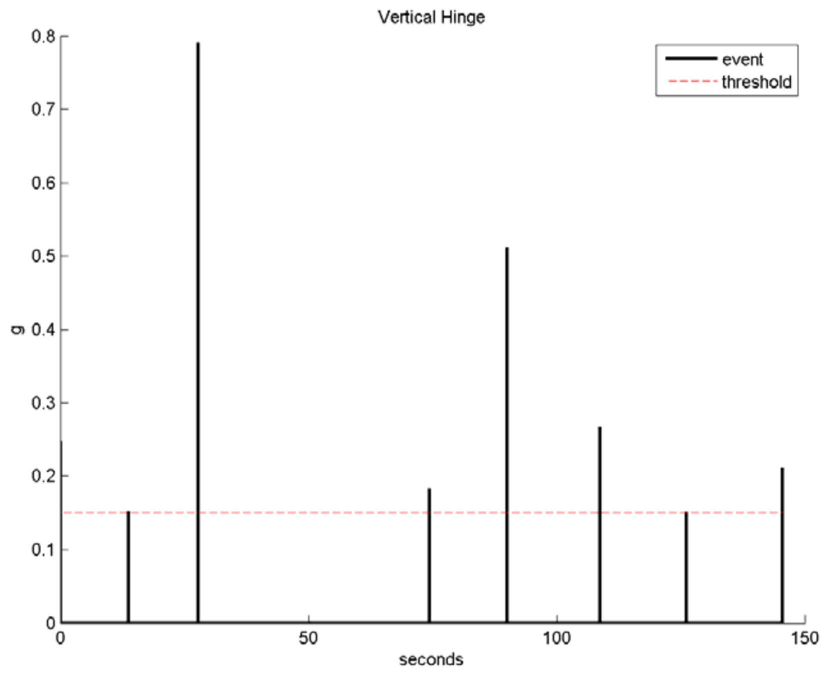


Figure 4.13 Vertical Hinge Event

4.2.3 External Events

The next group of episodic events can be grouped as external events. These sensors were fixed to a piece of furniture upon which the force of the human body was the key factor. The bump event would be attached to the bed frame, the bed event would hang from a small bag on the mattress, and the chair event had its sensor mounted to the back of a kitchen chair. These systems used motion to detect if a piece of furniture was interacted with. The results tend to show that they are dependent on the physique of the human subject. A larger subject would produce a stronger signal, while a smaller person would produce a much weaker signal. This can be seen in all three events.

The bed event in

Figure 4.14 and Figure 4.15 Bed event with second participant shows the difference between a large person and a small person getting onto the mattress while an accelerometer is attached. The heavier person displaces more of the mattress and the signals are much more pronounced.'

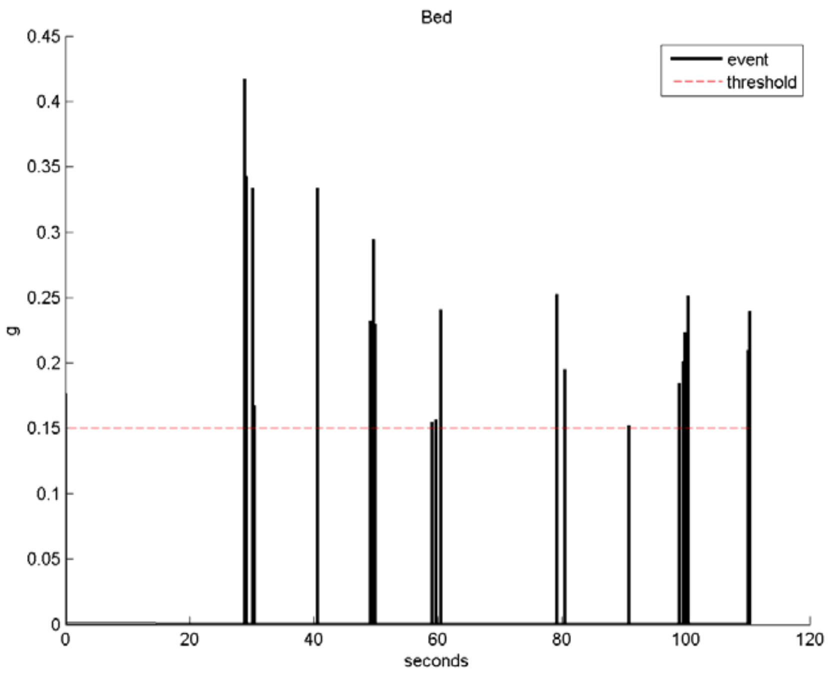


Figure 4.14 Bed Event for first participant

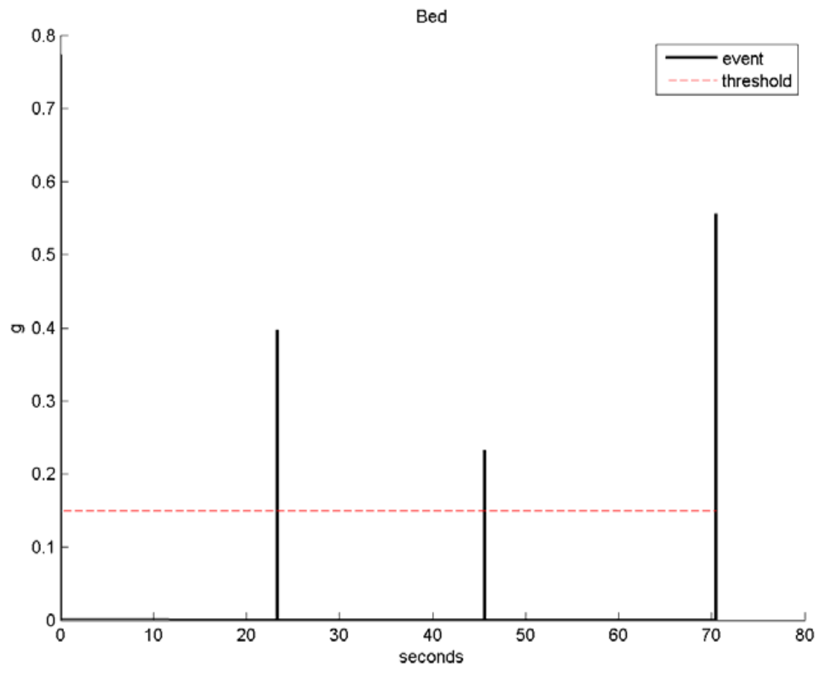


Figure 4.15 Bed event with second participant

The chair event in is much the same, since the chair is moved from under neath a kitchen table and then sat upon. A physically smaller person will not lift the chair high enough to trigger the y-axis of the accelerometer, causing almost no events to occur.

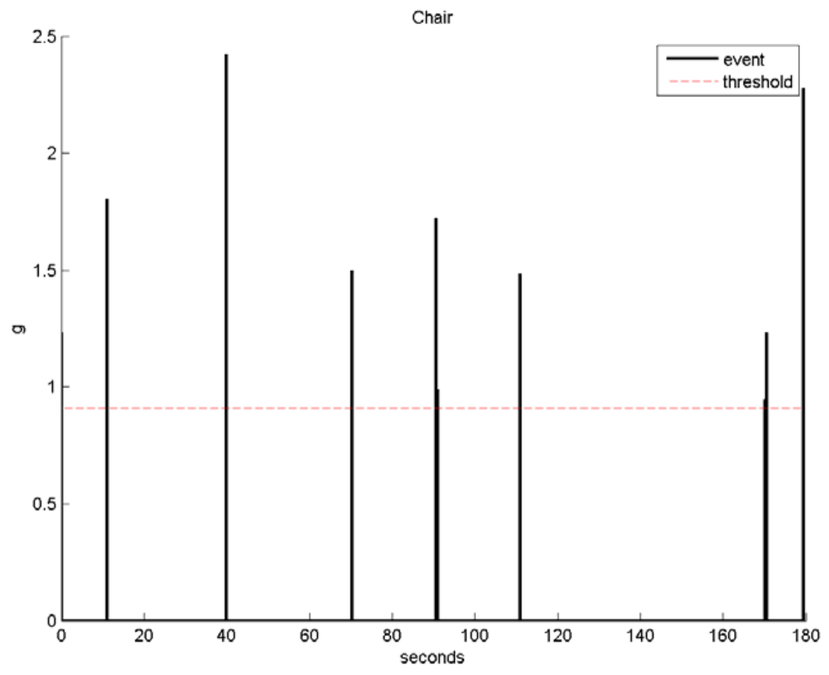


Figure 4.16 Chair Event with first participant

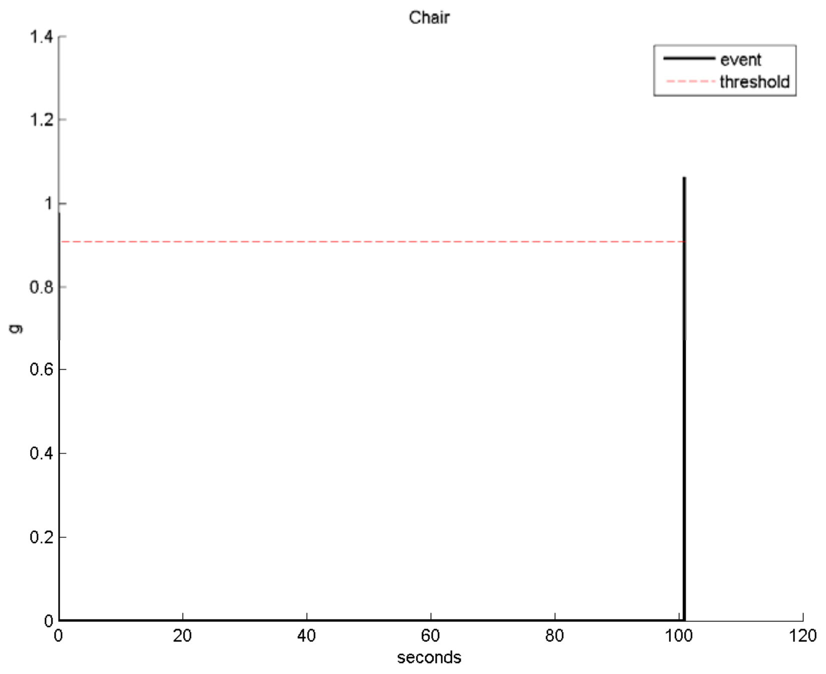


Figure 4.17 Chair event with second participant

Finally, bump event works well for the motion of its own self, but when attached to a piece of furniture, the amount of shock upon the furniture was inconsistent between different participants. However, when used independent of a piece of furniture, the bump sensor makes an excellent motion detector.

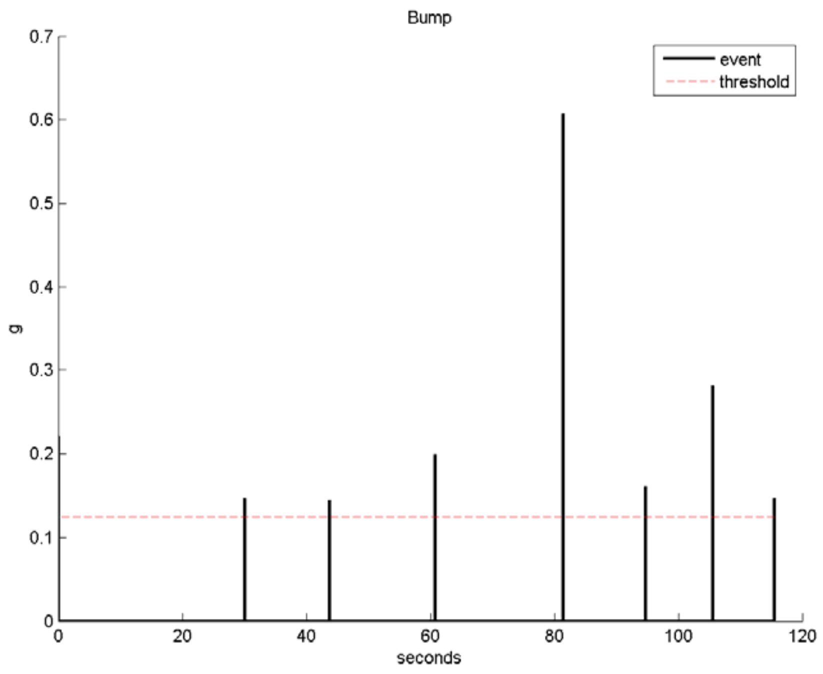


Figure 4.18 Bump Event with first participant

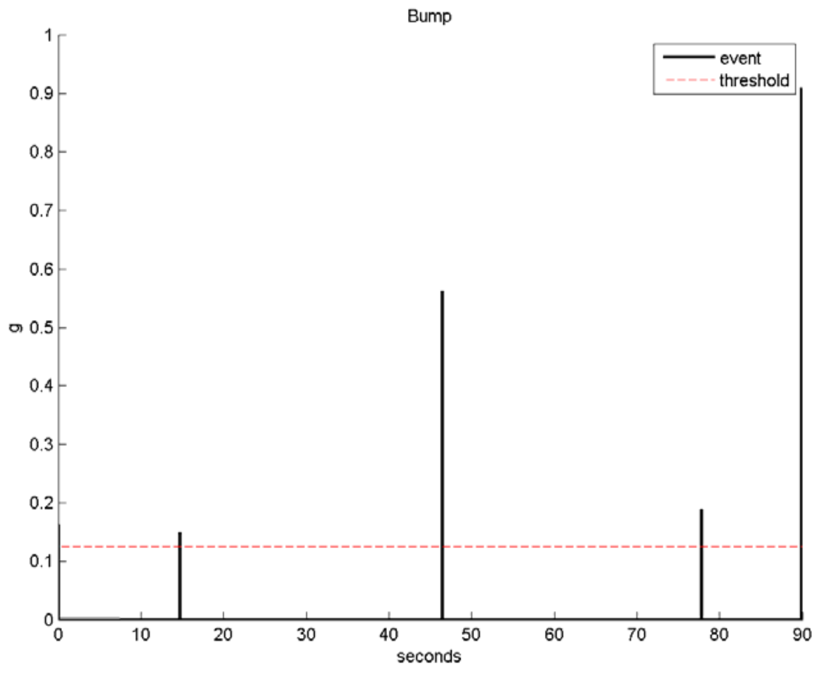


Figure 4.19 Bump event with second participant.

4.2.4 Worn Events

One type of event that is consistent with the effect of gravity is the response of the fall even t seen in Figure 4.20, where each case of the event occurred as one of the participants would drop to their knees to simulate a fall. Additionally, a sensor configured to recognize bump events can also be carried to show that the subject is currently in motion.

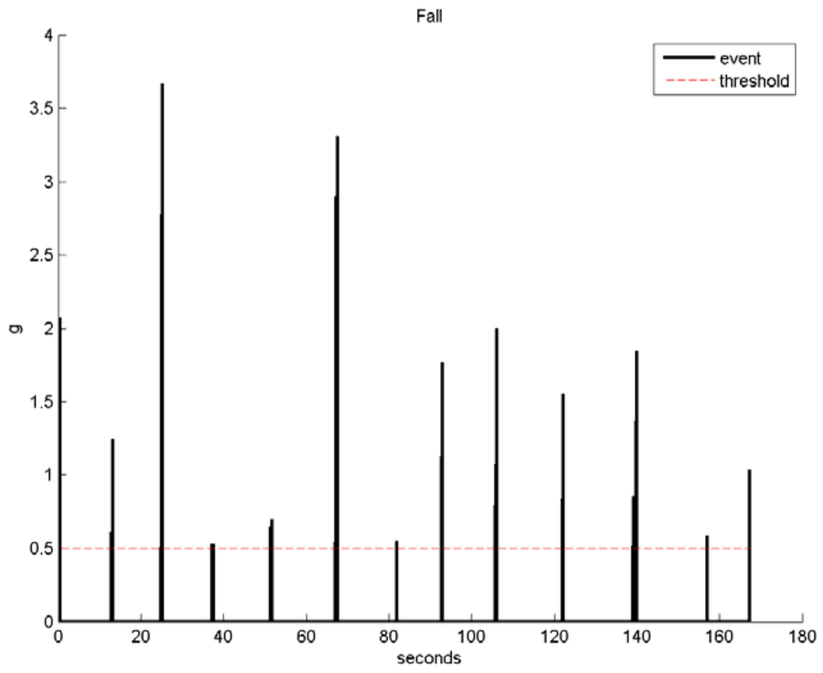


Figure 4.20 Fall Event

4.2.5 Comments on Event Validation

The behavior of the sensors during the validation experiments highlight which events are more reliable, and which need further study. The photocell events need to have further adaptation to make the sensor motes quickly deployable. The sensors using acceleration for a fixed range of motion do work, but like all radio signals are susceptible to obstacles within the environment. The sensors that are based on the whole human body acting externally on an object would have to be adjusted. Pressure pads and similar other sensors are more useful with someone sitting down on a piece of furniture. Modifications could be made to allow these kinds of sensors to feed their inputs into wireless sensor motes, and therefore enhance the episodic events. The worn sensors do seem to have a very regular and predictable behavior, and should be expanded to include both video verification as well as an expansion of the types of sensors worn. Perhaps a smartphone or other device with an accelerometer could be incorporated into the system.

4.3 Software Self-Repair

4.3.1 The Need for Self Repair

The need for self-repair comes from the concerns from when a software bug has a dramatic impact. The recent Toyota acceleration fault was not from replacement gas pedals, but has been admitted as being a software bug in the cruise control system.. [56] Several airline crashes, notably the flight of XYZ airbus, resulted from a loss of precision an onboard the inertial sensor. [57][58] In May, 2010, the NYSE suffered a blip when a software system reacted when the wrong price was offered on stocks. [59]

4.3.1.1 *Self-Adaptive*

The first step in a self-repairing software system is to first include a self-adaptive architecture to base the system upon. The planning required for a self-adaptive system has to be begun at the requirements level through the definition of key elements to define the requirements. In various different approaches, certain factors have to be defined first. For example, the requirements analysis has to include the ability to adapt as one of its key functions. Not only does the system have to be able to adapt, but the mechanisms needed to perform the adaptation have to be included. A system that has to be able to recompile its own code in order and redeploy must have a compiler program. Systems that are supposed to correct for power usage have to have power management. To do this, requirements have to include the mechanisms necessary to make the changes. Now, beyond the desire to adapt and the tools to make the adaptations, the changes themselves must be quantified. Ranges and needs have to be specified, as well as the consequences of the changes. In order to make a valid change to keep a software system adaptable to either its environment or to new technologies of the future, resources and parameters have to be defined. In addition to these desired changes, the desired consequences have to be considered. A predictability has to be chosen to make certain that adaptations have a value.[60][61][62]

4.3.1.2 *Self-Repair*

Self-adaptation, however, is the first stage of a self-repair system. The ability to recognize an event in the runtime of the program and to effect a change is the first need for the system. Now, the consideration is if the event detected is a runtime error in the software, requiring a dynamic change in the running of the system. Some parts of this method are an extension of the self-adapting software, and others include methods for the recognizing of errors in the code.

Different approaches are taken to dynamically improving software to overcome a runtime error. In some cases, a library of pre-generated object code is available, and the configuration of the running system.[63] In other cases that are not run-time, the known code can be parsed using a genetic algorithm to isolate cases of redundant or unreachable code. In some cases, such as the Microsoft Zune error, several code paths were generated, but only one held the true solution to the bug. [64]. Still other methods use grammars to compare the behavior of the code to the requirements of the program in order to find and correct the error. [65][66] In other systems, rather than to alter the code, individual components can be rebooted or re-assigned to another source.[67]

4.3.1.3 *Components of a Self Repairing System*

Once these requirements have been defined, it is necessary to look into the ways a real design can handle these projects. Two elements appear to be key in the research for developing a self-adaptive system. These elements are control and monitoring. Control is the component of the system that has to run a feedback loop that allows the processing space for the adaptations to be made. The monitoring is to be able to detect events within the system in order to trigger the evolution of the software.[68]

To begin with, a self-adaptive system must respond to changes in technology or the environment as time progresses. To do so, different types of monitoring mechanisms need to be put into place. Stubs, mirrors, gauges, wrappers and other objects are added to system code in order to track the behavior of the system.[69][70][71][62][72][73]. These inserted elements into the design must then relay their results back to a parallel process that can make decisions based on the detected events.

For the control mechanism, this tends to be a feedback loop, where the needs of the software follows the design goals of the system to see if the software is behaving in an acceptable manner.[74] Also the system has to be able affect the changes, whether it be the settings on a network relay or an actuator that can affect the environment. In addition, depending on the design, it is important to remember to include the human in the adaptation cycle. Computers should not be given the complete authority to self adapt in all situations without a human to take responsibility. The result of the control loop is to be that of a cyberphsyical feedback loop with a human being the judge of the adaptation. [75]

For monitoring, several different methods have been described but they all take a similar pattern. A stub is connected into the software at the point where the parameters defined in the requirements are at a key point. Multiple versions of these stubs are placed in key areas of the software architecture to report the state of the system to an external monitor of the running code. This external monitor can then report to the processes in the control loop to determine if an event is detected. [76] Another approach is to look for an alternative path of execution within the program rather than to throw an exception. [77]

The resulting pattern is the architecture of the software supports the ability to make changes through its life cycle. Changes can be made in runtime by monitoring the software and allowing the software to make self-configuration changes as necessary to keep the system functioning efficiently. [77]

4.3.2 Applying Self Repair to a WSN

Now, considering the definitions of the various self-repair systems, these methodologies could be applied to a wireless sensor network geared towards event detection in an assistive environment. The current WSN setup has event detection nodes deployed in the apartment and the results are transmitted back to a base station and a host machine that can act as a gateway to other media and servers. To follow the systems described, additional components would have to be included in the design of the system.

One system takes the approach of changing the WSN so instead of having a single base station sink, it has multiple base station sinks. These in turn are controlled by a single manager node that is the root now of all the base stations. This way, when a fault occurs on one of the base-stations, the manager node can reroute traffic to another base station. [79]

First, a sensor node does not, in itself, contain enough memory and computing power to include all the necessary analysis, code library, and compiling needed for a full self-repair system. The node, however, could report to the base stations that it is in need of repair, and even indicate in its message an error code that could be mapped to the exception handling available in java programming.

Second, the base station sink node may also develop a fault, as was noticed during the validation experiments. Having a single base station sink in the system is also a hindrance. Additionally, adding another layer of messaging and protocols to the WSN would include the loss of event detection messages in the system.

So, to improve the design of the WSN, a consideration is to include multiple base stations for event detection, and to add additional base stations for being the monitors of the self repairing aspect of the system.

An additional function added to these nodes is that instead of being on the same radio frequency as the event detection nodes, a node under repair would switch to a new frequency and connect to the gateway machine. The gateway machine would then update its local record of the WSN to show which nodes were undergoing repair, and which were still in standard operation. Now the system would operate on two frequencies, a repair frequency and an event detection frequency.

So, the system would start functioning, with the code stored in a library on the gateway machine, while the sensor nodes are in operation. When a runtime error occurs on a sensor node, the exception handling would be called. The sensor node would be in a recovery state, change its radio frequency, and signal one of the monitor base stations. The monitor would then report the area to the gateway computer, which would then begin to analyze the error in the code. The self-repair analysis could then occur on the gateway machine, reviewing the code and performing the analysis to solve the runtime error.

Once the update has been completed, then the system would perform an over-the-air (**OTA**) update of the sensor node. SunSPOTs support the ability to update their runtime image over the airwaves from an appropriate base station. Once the download has completed, the individual sensor nodes can restart and proceed to rejoin the wireless sensor networks and begin again event detection.

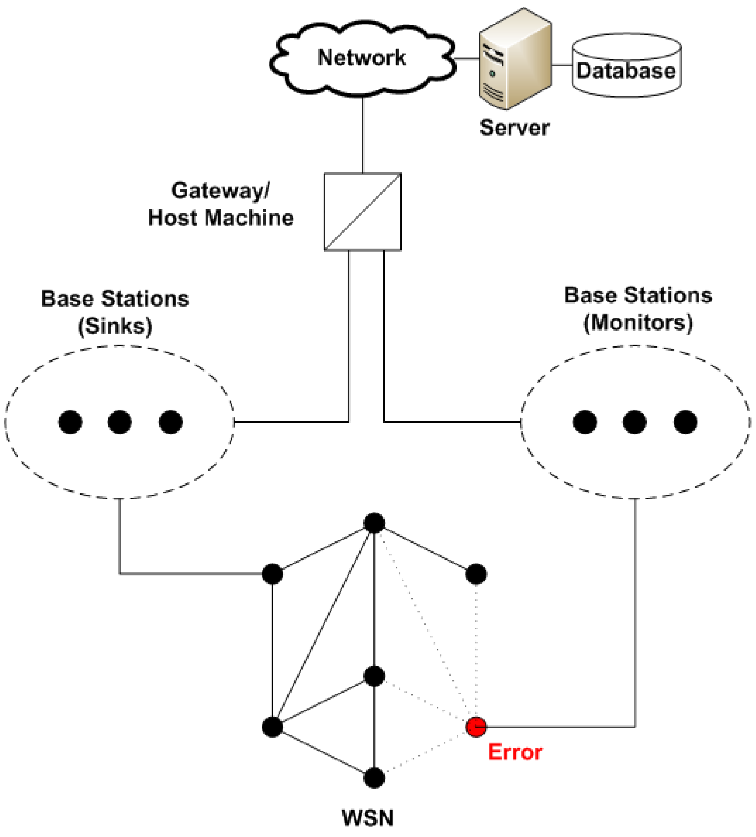


Figure 4.21 Possible Self Repair Framework for a WSN

4.4 The Role of the Human for Decision Making in a Cyberphysical Framework

The role of a human in a cyberphysical framework is not simply that the human is the user. There are different roles for the user depending on the tasks at hand when they interact with a cyberphysical framework. To that end, three key roles exist for the human: the subject, the expert, and the maintainer.

4.4.1 The User

The subject is the user, the person of interest, the patient in need of care. This role interacts with the cyberphysical framework in both the passive and active capacity. The various sensors and devices placed within the environment will observe and record the activity of the subject. In addition, the subject may receive messages from the framework to remind them of tasks, medications, or even emergency situations. Additionally, the subject may request information from the framework. After all, the information gathered is information about the subject.

4.4.2 The Caregiver

Depending on the nature of a cyberphysical feedback loop, someone has to have the knowledge or the skill to evaluate or make use of the data gathered on a daily basis. This daily user is the Expert, such as a nurse, technician, or other specialist who interacts with the system. If the framework makes a decision about the aforementioned subject, the resulting decision will sometimes need to be ratified by the expert. On other occasions, such as an event of interest occurring within the assistive environment, the Expert could be a stakeholder who should be contacted about the situation.

4.4.3 The Maintainer

The maintainer is the human who would look at the long term course of events in the assistive environment. The maintainer would be the human who could set up the various sensors and devices, be able to look into and configure the software, or even be a researcher who can request a log of the collected data for research.

4.4.4 Summary

The subject, the expert, and the maintainer are the three roles that a human fits into in this definition of a cyberphysical framework. An example is in the SmartDrawer project. In this project, we define the user, the caregiver, and the maintainer as three roles that interact with the system. The user is the subject, the design allowing them to enter prescriptions and set schedules. The caregiver is an expert, a nurse who would check the log of medication over the course of the day. And the maintainer would be able to check the system, and study long-term trends in the data.

CHAPTER 5

RELATED WORK

5.1 Introduction

Various other systems are in existence that utilize WSNs, RFID, smart furniture, and other devices. These systems go from individually worn sensor packages to complete installations at hospitals, nursing homes, or hidden in public view. Other methods exist for the placement of cameras or other sensors within a modeled 3D Environment.

5.2 Hospital Tracking Systems

A system using WSNs for medical environments is CodeBlue [80] where the authors propose a network infrastructure for emergency medical care. Code Blue is designed to be used with a single hospital and a mixture of WSN and different ad-hoc devices such as PDAs to display their results.[81]

Wireless sensor networks can have multiple uses from health care to security at a power station or for taking readings in a toxic environment where the motes have to be disposable.

5.3 Reminder Systems

A reminder system is a way of prompting a user, that they need to perform a certain task. This could be a cook in a kitchen, a manager at a warehouse, or a caregiver in an assistive environment. The Wisely Aware RFID Dosage (WARD) system is an integrated method of combining RFID information to ensure patient safety. RFID information from the patient is combined with the database of the medication to guarantee that medicine that could harm the patient due to allergy or other conditions is not dispensed. The goal is to remove any confusion of what medication a patient should receive using RFID tags to track their medical records[82]. Another system that uses prompting to remind a patient to do their needed exercise and take their medication is the AutoMinder system. The AutoMinder is a robotic assistant program that uses artificial intelligence to connect the needs of the patient with their schedule, and to prompt them with reminders of exercise, appointments, and medication by interfacing the patient's plan with various devices such as telephones and PDAs [83].

Another RFID system is the NAMA-RFID reminder system for keeping track of products. As inventory decreases due to the supply being tracked by RFID, then the user is prompted to remember to re-stock [84]. Such a similar event could be used to remind a user it is time to refill a prescription. Also, there are some other RFID applications are used in assistive environment. Such as Assistive Kitchen [85] from Technische Univ. in Germany, uses RFID tags on the objects in a kitchen, and later uses some mobile robot to sense the environment with RFID reader and also helps disabled people use such an assistive kitchen more efficiently. One of the more recent systems is called GlowCap [86], which is from Vitality, a startup company based in Cambridge, Massachusetts. Their reminder system targets the health and business problems with an Internet-connected bottle-cap. The GlowCap uses a wireless connection to report how the subjects take the medicine, as well as plays a tune to remind the subject when it is time to take their medicine. This reminder system also keeps track of the doses day by day using a commercial database. But the disadvantage for GlowCap is that they use a battery powered wireless connection, and GlowCap will be disabled when the battery charge is too low. If such a system were to use a passive RFID tagging system instead, the energy consumption would not be an issue.

5.4 Camera Systems

Cowan and Kovesi [87][88] propose a technique where spheres are placed tangent to the surfaces to be observed. They search the intersection of these spheres to find camera and lighting positions. The search of the volume of overlapped spheres is extended to consider a series of constraints to the camera field of view. Given that the camera is on a fixed distance from the target object, the length is adjusted to find the best clarity of the picture (number of pixels across) as well as how far away the camera must be in order to see multiple targets. They find the camera's position by minimizing and maximizing the length of the camera range according to these constraints.

Vásquez *at al.* [89] uses viewpoint entropy, a camera that sees from a single point of view in a 3D environment, to find how many surfaces can be seen from the sphere of view. These spherical approaches differ from our method by assuming a specific camera model that can see an entire sphere. Our method may consider any sensor model given a geometric solid to model its perception space.

Chen *at al.* [90] proposed an automatic camera placement system for the assembly of machines by robots. They modeled the environment and the components in three dimensions. The method finds the most efficient viewpoint for seeing all the components in the environment. The main goal is to find the fewest viewpoints in the environment that generates a complete picture of the situation. They set up a series of constrains on the image, as well as developing the Viewpoint Planner tool for handling multiple viewpoints.

Instead of using geometry, Fleishman *et al.* [91] used image-based modeling and rendered images to generate 3D scenes. Originally, cameras were placed into the scene using an ad-hoc method. The cameras are used to render and to view a 3D scene. However, their objective is to automatically place cameras to generate a dynamic scene from a *walking zone.* In this case, the view is constrained from the path of a 3D avatar. Ideally, the system should be able to view every polygon in the desired scene. The method tries to find the smallest number of cameras needed to complete the total view. This system could be used for a real world environment, by finding a subset of camera positions from a database of real camera positions.

Mason *et al.* [92] designed a prototyping tool for deploying sensors in a network, by taking a CAD model and finding multiple positions for the sensors. The system finds the precise points in 3D space necessary for testing a target object. These points are used to control the positioning of sensors around the surface of the object. In order to find the best possible position, special target markers are needed to increase the precision. In order to keep the operating costs low, the method tries to use as few images as possible.

Fiore *et al.* [93][94] addressed the problem of placing cameras in static positions to observe human activity. They developed a method of placing the cameras such that the chance of failure (of the algorithms used to observe activity and record events) is minimized. The algorithm searches in 5 degrees of freedom for placing the cameras to watch the area of interest. The behavior of the algorithm turned out to be non-linear in nature. The method is being enhanced to work with a new design for multiple moving cameras. In addition, the camera mounting is important to watch pedestrian motion and to enhance security. If the cameras are not mounted correctly, the gathering of the information will be inefficient and not suitable for handling the time and space constraints. The camera needs to be placed to see the key motions of the pedestrians and, consequently, requires a wide angle view. The system is working in real world conditions and is being expanded to work with camera pairs.

Bodor *et al.* [95] considered the problem of placing cameras to achieve a specific task. Unlike other methods, however, this system is using a series of images to analyze the 3D environment instead of using a 3D model. The goal of this research is to find the best way of placing the cameras in the environment to observe the scene. Traditionally, the approach is to place the cameras uniformly around the target to be observed, rather than take into consideration the human activity. The cameras are placed by analyzing the statistics of the motion pace to determine the scene. They consider moving, fixed, and occluded scenes while using real world time constraints.

David *et al.* [96] developed the CAPTHOM system, a method of finding the human presence within an active environment. The goal was to build a system for the placement of the cameras that could both monitor the elderly and be energy efficient. They built a working simulator and a software tool to plan the best location for the sensors. The best method for installing the sensor network consisted of establishing the objectives, methods of detection, and clearly defining the problems in the system. The method employs two models, the model of the sensor to be placed in the environment, and the model of the environment itself. The constraints placed on the system depend on where the sensor views overlap, how many sensors are needed, and the points of electrical consumption within the environment. Currently, the scenes are for empty rooms only, but the authors plan to expand the system to include rooms populated with furniture.

He *at al.* [97] studied various forms of localization for geographic placement of sensors. This is more of a 2D approach than a 3D approach. They overlap target areas and examine DV-Hop Counting propagation as a means to find the locations of the various sensors, rather than finding where to place the various sensors.

5.5 Behavior Detection Systems

Other universities and research groups are also studying human activity in assistive environments. Different groups have used various combinations of sensors and methods to study aspects of human behavior. Also, previous work related to this topic is also discussed.

Lymberopoulos et. al. propose a methodology to extract human being's activity from a person who is living in a home equipped with wireless sensor networks [98]. They use various types of sensors such as tracking cameras, door sensors, and passive infrared sensors (PIR) to monitor the person's behavior. Using the raw data of the behavior, a sequence of detected sensing features is produced over time. They consider spatiotemporal feature, which has time, duration, and location. By these three elements, they can interpret the situation better, or even the same event in a totally different way. In order to extract daily activity patterns from data sequences, they use a-priori algorithm [15], which tries to find all the frequent sequences by searching the input set of episodes exhaustively. By this way, they try to find most frequent spatiotemporal activity patterns. Experiments are done for an elderly person in a home with PIR sensors for 30 days to extract daily activity patterns.

Wood et. al, propose a framework of context-aware wireless sensor networks for assisted living and residential monitoring in [100]. The name of the system, AlarmNet is for elderly residents or patients who are needed to be monitored continuously in terms of their accidental behavior or health condition change. This system uses wireless sensor devices worn by patients such as ECG, pulse oximeter, or accelerometers, and emplaced sensors to measure or detect dust, temperature, or resident activities. SATIRE [101] is used for body networks that have wireless sensor devices to classify activities of daily living by analyzing data from accelerometer worn by residents or patients. SATIRE uses HMM (Hidden Markov Model) to identify human being's activity. In order to give ad hoc queries and get answers from the stored sensed data, they also developed SenQ [102], which is a query management system in AlarmNet. Context-aware power saving for wireless sensors is adopted by turning off unnecessary sensors by considering current situation with spatial and temporal information.

Tapia et. al, build a system for experiment to recognize human being's activities using low-cost state-change sensors in their paper [103]. They install the sensors in two one-bedroom apartments so that they can gather all the activities from residents who are living in the apartment individually. Since many people do not like to be monitored especially by camera, they develop simple state-change sensors instead of using multi-purpose sensor motes. They are interested in recognizing activities such as grooming, cooking, and toiletry, that can be detected by the sensors they developed. Initially, they get pre-knowledge about what the residents are doing at particular moment by the context-aware experience sampling tool (ESM) so that the information can be used as a training set. Then the system calculates the temporal features (exists and before) to generate training examples. Bayesian network classifier is used to calculate the probability of the current activity. The work emphasized on user friendly sensors that mitigate the invasion of privacy when data for activities are collected.

CHAPTER 6

SUMMARY AND EXTENSIONS

6.1 Summary

The original goal of the research was to establish a testbed apartment in which algorithms and methods could be tried and tested. True, this basic component was accomplished using the various brands of sensors, both wireless and wired, that have been deployed in the assistive living apartment. In addition, other components such as the automatic placement of sensors or the smart drawer have been included during the testing of the gear and the construction and reconstruction of the apartment. Wireless sensor networks, 3D modeling, radio frequency identification, and wired sensors have been combined with smart furniture, worn sensors, and verification cameras to result in a series of experiments that allow the monitoring of human behavior in a non-intrusive manner.

6.2 Extensions

The most obvious extensions of this work include the handling and processing of the episodic events as they are transmitted by the system and then stored for later analysis. Various different methods can be used in order to process the episodic events and events of interest to look for trends in human behavior or for abnormalities or for sudden changes.

6.2.1 Applications for the Assistive Apartment.

Once the assistive apartment has been constructed, and the wireless sensor network and other devices have been deployed, various methods can be applied to look for human behavior in the environment. One method of looking for abnormalities is to establish a dictionary of known activity, based on a series of events. Once the episodic events are constructed into a series, they can be processed to remove duplicates or even recognize events out of order. Once the sequences are cleaned and ready for analysis, the resulting comparison of a dictionary of known activity can be checked against the current actions within the apartment to look for unknowns that signal a change in behavior.[54] Another approach is to use a Partial Order Markov Decision Process, which takes a combination of the episodic events as one of the inputs into the decision making process. Various inputs into the system allows the system to maximize the reward for a function that detects the state of the human in the environment. When the conditions are needed, some action, such as an internet phone call for help, can be made by the system. [93] Yet another approach is the application of Hidden Markov Models(**HMM**). A HMM can be used to translate or map one series of inputs or observations into another set of labels that would identify the current action of the participant in the assistive apartment. [105][106][96][108]. All of these routines can be used to process the episodic events into responses of human activity.

6.2.2 Advanced Approaches for the Assistive Living Apartment.

Two other methods are of interest for the cyberphysical feedback loop and the scholastic context free grammars, which extend beyond simply applying and analyzing the events.

For the cyberphysical feedback loop, there are two halves of the cycle to consider. The first half of the loop includes the episodes coming from the apartment and being available for a model or other refinement algorithm. In the ideal case, the system would recognize the activity, and proceed to use actuators to make physical changes to the assistive environment. [7] Simple approaches to such a system would be to control the use of the power saving properties of the sensors deployed in the environment. A second layer of control would be to include actuators able to move and adjust, or be able to recommend changes in the layout of the apartment to improve the quality of life of the inhabitant. Additional components could include smart energy appliances, that would be able A final layer would be the deployment of robots that could respond to the conditions in the apartment, and able to handle either emergencies or even respond to conditions such as being a reminder for the taking of medication, exercise, or appointments.

For the second advanced study, would include the use of stochastic context free grammars (SCFG) for the manipulation of the data in a hierarchal, rule based way that would be able to handle multiple possible combinations of events in order to identify a behavior. In a SCFG, as each token is processed, one or more rules are tracked and assigned a probability. When the probability becomes too low, that branch of the grammar is pruned and the process continues. [109] The research by Lymberopoulos, *et al.* [110], took the movement patterns from a camera view, and constructed a hierarchal SCFG based upon the direction of change of the individual. This method had the advantage of Another application of the SCFG turn was based on the work of Ogale [111] which includes the recognizing of the position of a human figure using the Natural Languages Toolkit. The toolkit breaks down the various data stream observations into 'phonemes' that can be used to construct the SCFG. The problem with these methods is that a series of phonemes can occur in a series, and this can be solved by merging the events together and changing them from being a single time stamp to a start time and a duration, such as the chunking method by Geyik.[112]

6.3 Future Work

The future work for this framework of tools and methods for human activity recognition is just the beginning of a testbed on which multiple projects can be continued. New technologies and other actions within the apartment testbed can lead to a variety of new research areas. Beyond the devices in the current version of the testbed, smart energy meters and appliances can have their own data streams that can be interpreted into events. The smart furniture can be expanded with more sensors and the WSN could be adapted to run some of the wired sensors from the Phidget family. The apartment testbed could be used to train and run robots within the environment to be of assistance to participants either as agents to remind them of their medicine, exercise, or therapy. In addition, they can be used as a remote platform for a caretaker to investigate the apartment in case of emergency or accident. Beyond these physical aspects, the data streams from the various inputs can be routed through a framework that can be fitted with metrics for database tagging or for quality of service measurements and experiments. And anything dealing with the private information about the activities of the individual provides a strong reason to investigate the security and privacy concerns these types of technology will generate. And all of these future projects would need to be controlled by a human operator, requiring research into fast, easy to use interfaces so that both the individual, or a care taker, or a researcher could find the information they need in time to be useful.

REFERENCES

[1] 2007 "When Everything Connects: Information technology has nothing to lose but its cables" The Economist April 28, pp 1

[2] Akyildiz, I., Su, W, Sankarasubramaniam, Y., Cayirci, E., (2002)A survey on sensor networks. Communications Magazine, IEEE, 2002 [S]

[3] Vieira, M, Coelho, C., da Silva D, daMata , J (2003) Survey on wireless sensor network devices. IEEE Emerging Technologies and Factory Automation, 2003

[4] Vieira, M, Coelho, C., da Silva D, daMata , J (2003) Survey on wireless sensor network devices. IEEE Emerging Technologies and Factory Automation, 2003

[5] http://polymnia.pc.unicatt.it

[6] D.I. Kosmopolous, A. Doulamis, A.Makris, N.Doulamis, S.Chatzis, S.E. Middleton, "Vision-based production of personailzed Video" Signal Processing: Image Communication 24, 2009 p 158-176.

[7] F. Makedon, Z. Le, H. Huang, E. Becker, and D. Kosmopoulos, "An Event Driven Framework For Assistive Cps Environments." SIGBED 2009

[8] Merriam-Webster. Merriam-Webster's Collegiate Dictionary, 11th Edition thumb-notched with Win/Mac CD-ROM and Online Subscription. Merriam-Webster, July 2003

[9] I. Sinclair, Sensors and Transducers, Newnes, London, 3rd ed. , 2001

[10] j. Wilson, Sensor Technology Handbook, Elsevier Inc, 2005

[11] Carr Sensors and Circuits: Sensors, Transducers, and Supporting Circuits for Electronic Instrumentation, Measurement, and Control

[12] ADT7411 Data Sheet

[13] Phidget Data Sheet #1018

[14] K. Finkenzeller, RFID Handbook: Radio-frequency Identification Fundamentals and Applications, John Wiley, 1999.

[15] http://www.asihome.com/ASIshop/product_info.php?products_id=735

[16] http://www.unitedsecurity.com/pages/pressure.html

[17] “Vista Medical” http://www.pressuremapping.com/

[18] Brett Warneke, Matt Last, Brian Liebowitz, Kristofer S.J. Pister, "Smart Dust: Communicating with a Cubic-Millimeter Computer," Computer, pp. 44-51, January, 2001

[19] Moteiv Inc. Invent Motes. http://www.moteiv.com

[20] www.sunspotworld.com/

[21] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, “ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring,” University of Virginia Computer Science Department Technical Report, 2006.

[22] Eric Becker, Yurong Xu, Steven Ledford, Fillia Makedon, “A wireless sensor network architecture and its application in an assistive environment,” Proceedings of the 1st International Conference on Pervasive Technologies Related to Assistive Environments, PETRA 2008: 25

[23] Eric Becker, Zhengyi Le, Kyungseo Park, Yong Lin, and Fillia Makedon, “Event□based Experiments in an Assistive Environment using Wireless Sensor Networks and Voice Recognition,” Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'09), Corfu, Greece, June 9□13, 2009

[24] Eric Becker, Roman Arora, Scott Phan, Jyothi Vinjumer, Fillia Makedon , Extending Event-Driven Experiments for Human Activity for an Assistive Environment Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'10), Corfu, Greece, June 23□25, 2010.(Accepted)

[25] Doukas, C. and Maglogiannis, I. 2008. Enabling human status awareness in assistive environments based on advanced sound and motion data classification. In Proceedings of the 1st international Conference on Pervasive Technologies Related To Assistive Environments (Athens, Greece, July 16 - 18, 2008). F. Makedon, L. Baillie, G. Pantziou,

and I. Maglogiannis, Eds. PETRA '08, vol. 282. ACM, New York, NY, 1-8. DOI= http://doi.acm.org/10.1145/1389586.1389588

[23] Becker, E., Le, Z., Park, K., Lin, Y., and Makedon, F. 2009. Event-based experiments in an assistive environment using wireless sensor networks and voice recognition. In Proceedings of the 2nd international Conference on Pervsive Technologies Related To Assistive Environments (Corfu, Greece, June 09 - 13, 2009). PETRA '09. ACM, New York, NY, 1-8. DOI= http://doi.acm.org/10.1145/1579114.1579131

[27] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In Proceedings of BSN '09, pages 138--143, Washington, DC, USA, 2009.

[28] sketchup.google.com

[29] www.blender.org

[30] Hess, R. 2007 The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. No Starch Press.

[31] TinyOS www.tinyos.net

[32] http://bellard.org/tcc/

[33] Moteiv Inc. Invent Motes. http://www.moteiv.com

[34] Sun Spot, Sun MicroSystems, http://www.sunspotworld.com/

[35] ”Phidgets Inc. - Unique and Easy to Use USB Interfaces.” http://www.phidgets.com

[36] B.S. Ashar and A. Ferriter, “Radiofrequency Identification Technology in Health Care: Benefits and Potential Risks,” JAMA, vol. 298, 2007, p. 2305.

[37] A. Cangialosi, J.E. Monaly Jr, and S.C. Yang, “LEVERAGING RFID IN HOSPITALS: PATIENT LIFE CYCLE AND MOBILITY PERSPECTIVES,” Communications Magazine, IEEE, vol. 45, 2007, pp. 18-23.

[38] D. Young, “Pittsburgh hospital combines RFID, bar codes to improve safety.,” American Journal of Health-System Pharmacy, vol. 63, 2006, p. 2431.

[39] . Havenstein, “Wholesaler Set to Use RFID to Track Drugs,” Computerworld , vol. 40, 2006, p. 17.

[40] “Medication Tracking System Gets Under Way,” Pharmacy Times , vol. 73, p. 50.

[41] B. Brewin, “FDA backs RFID tags for tracking prescription drugs,” Computerworld, vol. 38, 2004, p. 4.

[42] E. Malykhina, “DRUGMAKER SHIPS RFID TAGS WITH OXYCONTIN,” Information Week, p. 20.

[43] Yurong Xu, Joshua Katuri, Eric Becker, Fillia Makedon, “RFID□based Smart Medicine Drawer for Assistive Environments,” BIOT’08

[44] Eric Becker, Vangelis Metsis, Roman Arora, Jyothi Vinjumur, Yurong Xu, Fillia Makedon, “SmartDrawer: RFID□Based Smart Medicine Drawer for Assistive Environments,” Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'09), Corfu, Greece, June 9□13, 2009.

[45] J.M. Mallion, C. Dutrey-Dupagne, L. Vaur, N. Genes, M. Renault, F. Elkik, P. Baguet, and S. Boutelant, “Benefits of electronic pillboxes in evaluating treatment compliance of patients with mild to moderate hypertension,” Journal of hypertension, vol. 14, 1996, pp. 137–144.

[46] Konstantinides, D.G., “Risk Models with Extremal Subexponentiality.” Brazilian Journal of Probability and Statistics,21,No.1,63-84,2007.

[47] R. Rajagopalan and P.K. Varshney, “Connectivity analysis of wireless sensor networks with regular topologies in the presence of channel fading,” IEEE Transactions on Wireless Communications, vol. 8, 2009, pp. 3475–3483.

[48] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” Computer Networks, vol. 52, 2008, pp. 2292–2330.

[49] Y.P. Kim, E. Jung, and Y.J. Park, “A Radio-Aware Routing Algorithm for Reliable Directed Diffusion in Lossy Wireless Sensor Networks,” Sensors, vol. 9, 2009, p. 8047.

[50] H. Pishro-Nik, K. Chan, and F. Fekri, “Connectivity properties of large-scale sensor networks,” Wireless Networks, vol. 15, 2009, pp. 945–964.

[51] H. Hu and Q. Zhu, “Power control based cooperative opportunistic routing in wireless sensor networks,” Journal of Electronics (China), vol. 26, 2009, pp. 52–63.

[52] M. Haenggi, J.G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, “Stochastic geometry and random graphs for the analysis and design of wireless networks,” IEEE Journal on Selected Areas in Communications (to appear), 2009

[53] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (Mobi-COM), August 2007.

[54] K. Park, E. Becker, J.K. Vinjumur, Z. Le, and F. Makedon, “Human behavioral detection and data cleaning in assisted living environment using wireless sensor networks,” Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments, Corfu, Greece: ACM, 2009

[55] Chiang, M.-W.; Zilic, Z.; Radecka, K.; Chenard, J.-S., "Architectures of increased availability wireless sensor network nodes," in Proceedings of the International Test Conference (ITC 2004),pp.1232-1241,Oct.2004

[56] “Toyota Announces Voluntary Recall on 2010 Model-Year Prius and 2010 Lexus HS 250h Vehicles to Update ABS Software”, www.toyota.com

[57] Taylor, Rob “Computer Glitch may have caused Qantas Jet Plunge” Retuers, Oct 8, 2008

[58] Iverson, Jeffrey, “Could a Computer Glitch have Brought Down Air France 447”, Time June 5, 2009

[59] Memmont, Mark, “Stock Market's Nosedive Wasn't Set Off By A Trader's Typo, 'Journal' Says”, National Public Radio, May 11, 2010

[60] B. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, and others, “Software engineering for self-adaptive systems: A research roadmap,” Software Engineering for Self-Adaptive Systems, 2009, pp. 1–26.

[60] Andersson, J., de Lemos, R., Malek, S., Weyns, D.: Towards a classification of self-adaptive software system. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525. Springer, Heidelberg (2009)

[61] J. Andersson, R. de Lemos, S. Malek, and D. Weyns, “Reflecting on self-adaptive software systems,” 2009.

[62] D. Garlan and B. Schmerl, “Model-based adaptation for self-healing systems,” Proceedings of the first workshop on Self-healing systems, 2002, p. 32.

[63] G. Edwards, J. Garcia, H. Tajalli, D. Popescu, N. Medvidovic, G. Sukhatme, and B. Petrus, “Architecture-Driven Self-Adaptation and Self-Management in Robotics Systems,” Proceedings of SEAMS, 2009

[64] S. Forrest, T.V. Nguyen, W. Weimer, and C. Le Goues, “A genetic programming approach to automated software repair,” Proceedings of the 11th Annual conference on Genetic and evolutionary computation, 2009, pp. 947–954.

[65] A. Bucchiarone, P. Pelliccione, C. Vattani, and O. Runge, “Self-Repairing systems modeling and verification using AGG,” 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, Cambridge, United Kingdom: 2009, pp. 181-190.

[66] H. Ehrig, C. Ermel, O. Runge, A. Bucchiarone, and P. Pelliccione, “Formal Analysis and Verification of Self-Healing Systems,” Fundamental Approaches to Software Engineering, pp. 139–153.

[67] F. Boyer, N. De Palma, O. Gruber, S. Sicard, and J.B. Stefani, “A Self-repair Architecture for Cluster Systems,” Architecting Dependable Systems VI, 2009, pp. 124–147.

[68] O. Derin, A. Ferrante, and A.V. Taddeo, “Coordinated management of hardware and software self-adaptivity,” Journal of Systems Architecture, vol. 55, 2009, pp. 170–179.

[69] A. van Hoorn, W. Hasselbring, and M. Rohr, “Engineering and Continuously Operating Self-Adaptive Software Systems: Required Design Decisions,” 2009.

[70] S.W. Cheng, V. Poladian, D. Garlan, and B. Schmerl, “Improving Architecture-Based Self-Adaptation through Resource Prediction,” Software Engineering for Self-Adaptive Systems, 2009, pp. 71–88.

[71] J .H. An, M.E. Shin "Self-reconfiguration in self-healing systems," 2008. Proceedings of the Third IEEE International Workshop on Enginneering of Autonomic and Atonomous Systems (EASE'06)

[72] D. Garlan, S.W. Cheng, and B. Schmerl, "Increasing system dependability through architecture-based self-repair," Architecting Dependable Systems, 2003, pp. 61–89.

[73] S.W. Cheng, D. Garlan, B. Schmerl, J.P. Sousa, B. Spitznagel, and P. Steenkiste, "Using architectural style as a basis for system self-repair," Proceedings of the IFIP 17th World Computer Congress-TC2 Stream/3rd IEEE/IFIP Conference on Software Architecture: System Design, Development and Maintenance, 2002, pp. 45–59.

[74] A. van Hoorn, M. Rohr, A. Gul, and W. Hasselbring, "An adaptation framework enabling resource-efficient operation of software systems," International Conference on Software Engineering 2009, 2009, pp. 0–3.

[75] Y. Brun, G. Di Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. M\üller, M. Pezzè, and M. Shaw, "Engineering self-adaptive systems through feedback loops," Software Engineering for Self-Adaptive Systems, 2009, pp. 48–70.

[76] M.M. Al-Zawi, A. Hussain, D. Al-Jumeily, and A. Taleb-Bendiab, "Using Adaptive Neural Networks in Self-Healing Systems," 2009 Second International Conference on Developments in eSystems Engineering, 2009, pp. 227–232.

[77] A. Gorla, L. Mariani, F. Pastore, M. Pezzè, and J. Wuttke, "ACHIEVING COST-EFFECTIVE SOFTWARE RELIABILITY THROUGH SELF-HEALING," Computing and Informatics, vol. 2, 2010, pp. 1001–1022.

[78] Y. Wang and J. Mylopoulos, "Self-Repair through Reconfiguration: A Requirements Engineering Approach," 2009 IEEE/ACM International Conference on Automated Software Engineering, 2009, pp. 257–268.

[79] G. Yoo and E. Lee, "Self-Healing Methodology in Ubiquitous Sensor Network," Self, vol. 3, 2009

[80] Malan, D., Fulford-Jones, T., Welsh, M., Moulton, S. Code Blue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. Workshop on Wearable and Implantable Body Sensor Networks, 2004

[81] Code Blue Wireless Sensor Networks for Medical Care. http://www.eecs.harvard.edu/~mdw/proj/codeblue/

[82] P.R. Sun, B.H. Wang, and F. Wu, “A New Method to Guard Inpatient Medication Safety by the Implementation of RFID,” Journal of Medical Systems, vol. 32, 2008, pp. 327-332.

[83] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, “Towards robotic assistants in nursing homes: Challenges and results,” Robotics and Autonomous Systems, vol. 42, 2003, pp. 271-281.

[84] O. Kwon and S. Choi, “Applying associative theory to need awareness for personalized reminder system,” Expert Systems With Applications, vol. 34, 2008, pp. 1642-1650.

[85] M. Beetz, J. Bandouch, A. Kirsch, A. Maldonado, A. Muller, and R.B. Rusu, “The Assistive Kitchen—A Demonstration Scenario for Cognitive Technical Systems 0,” Proceedings of the 4th COE Workshop on Human Adaptive Mechatronics (HAM), 2007.

[86] M. Copeland, “Vital medicine,” FORTUNE MAGAZINE, 2008.

[87] C. Cowan and P. Kovesi, “Automatic sensor placement from vision task requirements,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 3, 1988, pp. 407-416.

[88] C. Cowan, “Automatic camera and light-source placement using CAD models,” In Proceeding of the Workshop on Directions in Automated CAD-Based Vision, 1991, pp. 22-32.

[89] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Automatic view selection using viewpoint entropy and its application to image-based modelling,” Computer Graphics Forum, vol. 22, no. 4, 2004, pp. 689-700.

[90] S. Chen and Y. Li, “Automatic sensor placement for model-based robot vision,” IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 34, no. 1, 2004, pp. 393-408.

[91] S. Fleishman, D. Cohen-Or, and D. Lischinski, “Automatic camera placement for image-based modeling,” In Proceedings of the Pacific Conference on Computer Graphics and Applications, 1999, pp. 12-20.

[92] S. Mason and A. Grün, “Automatic sensor placement for accurate dimensional inspection,” Computer Vision and Image Understanding, vol. 61, no. 3, 1995, pp. 454-467.

[93] L. Fiore, D. Fehr, R. Bodor, A. Drenner, G. Somasundaram, and N. Papanikolopoulos, “Multi-camera human activity monitoring,” Journal of Intelligent and Robotic Systems, vol. 52, no. 1, 2008, pp. 5-43.

[94] L. Fiore, G. Somasundaram, A. Drenner, and N. Papanikolopoulos, “Optimal camera placement with adaptation to dynamic scenes,” In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2008, pp. 956-961.

[95] R. Bodor, A. Drenner, P. Schrater, and N. Papanikolopoulos, “Optimal camera placement for automated surveillance tasks,” Journal of Intelligent and Robotic Systems, vol. 50, no. 3, 2007, pp. 257-295.

[96] P. David, V. Idasiak, and F. Kratz, “A sensor placement approach for the monitoring of indoor scenes,” Lecture Notes in Computer Science, vol. 4793, 2007, pp. 110-125.

[97] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in Proceedings of the Annual International Conference on Mobile Computing and Networking, 2003, pp. 81-95.

[98] D. Lymberopoulos, A. Bamis, and A. Savvides, Extracting spatiotemporal human activity patterns in assisted living using a home sensor network, In PETRA ’08, pages 1–8, 2008.

[99] R. Agrawal and R. Srikant, Fast Algorithm for Mining Association Rules in Large Databases, In VLDB ’94, pages 487-499.

[100] A. D. Wood, J. A. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru, Context-aware wireless sensor networks for assisted living and residential monitoring, IEEE Network, 22(4):26–33, 2008.

[101] R. K. Ganti, P. Jayachandran, T. F. Abdelzaher, and J. A. Stankovic, SATIRE: A Software Architecture for Smart AtTIRE, In Mobisys, 2006

[102] A. D. Wood, L. Selavo, and J. A. Stankovic, SenQ: An Embedded Query System for Streaming Data in Heterogeneous Interactive Wireless Sensor Networks, In DCOSS, 2008

[103] E. M. Tapia, S. S. Intille, and K. Larson, Activity recognition in the home using simple and ubiquitous sensors, In Pervasive Computing, pages 158–175, 2004.

[104] Y. Lin, E. Becker, K. Park, Z. Le, and F. Makedon, “Decision making in assistive environments using multimodal observations,” Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments, 2009, p. 6.

[105] L.R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, vol. 77, 1989, pp. 257-286.

[106] L. Rabiner and B. Juang, “An introduction to hidden Markov models,” ieee assp magazine, vol. 3, 1986, pp. 4-16.

[107] “Jahmm - Project Hosting on Google Code.” http://code.google.com/p/jahmm/

[108] Eric Becker, Roman Arora, Scott Phan, Jyothi Vinjumer, Fillia Makedon, "Extending Event-Driven Experiments for Human Activity for an Assistive Environment", Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'10), Samos, Greece, June 23□25,

[109] D. Moore and I. Essa, “Recognizing multitasked activities from video using stochastic context-free grammar,” Proceedings of the National Conference on Artificial Intelligence, 2002, pp. 770–776.

[110] D. Lymberopoulos, A.S. Ogale, A. Savvides, and Y. Aloimonos, “A sensory grammar for inferring behaviors in sensor networks,” Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on, 2006, pp. 251–259.

[111] A. Ogale, A. Karapurkar, and Y. Aloimonos, “View-invariant modeling and recognition of human actions using grammars,” Dynamical Vision, pp. 115–126.

[112] S.C. Geyik and B.K. Szymanski, “A Grammar Inference Algorithm for Event Recognition in Sensor Networks.”

BIOGRAPHICAL INFORMATION

Eric Becker received his Bachelor of Science in Computer Science and Engineering from the University of Texas at Arlington in 1995, and later received his Master of Science in Computer Science and Engineering in 2000. His research interest include embedded computing, software and systems engineering, automation of health care, and assistive environments. Other interests include symbolic execution, compiler theory, and real-world acquisition of data in real time. Eric Becker's past and ongoing projects include the Object Oriented Toolkit and related projects while he was with the Software Engineering Center for Telecommunications.. While working with the Transportation Instrumentation Laboratory, projects include the Road Profiler, the Sliding Profiler, the Cross-slope project and more systems for real-time data acquisition in association with the Texas Department of Transportation and Texas A&M University. With the Heracleia Human Centered Computing Laboratory, projects include the Assistive Living Apartment WSN, the Smart Drawer project, the Automatic Sensor Placement Project as well as participating in related research. Future projects include fall detection with computer vision, smart energy with the CONVIA systems, and virtual reality. Future plans include seeking a post doctorate abroad, and seeking an academic position.