An Interactive Visualization Toolkit for Exploring Ambient-Assistive Living Traces

by

Shahrzad Rafatnia

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

The Smart-Condo™ project investigates how technology can be used to unobtrusively observe and support seniors to live independently in their homes longer. Sensors embedded in the environment react to the people's daily-living activities and the recorded readings are analyzed to recognize these activities, as well as to infer whether they are typical or exceptional. In this thesis, we present a visualization system for analyzing and exploring the sensor data recorded in the Smart Condo, and more generally "smart homes". The purpose of the system is to support caregivers in easily perceiving the activities of the Smart-Condo™ occupants; therefore, our toolkit offers a variety of visualizations, including spatio-temporal dynamic activity views and aggregate activity statistics graphs.

Acknowledgements

I would like to express my deep gratitude to my supervisor Professor Eleni Stroulia for her supervision, guidance and great patience throughout this research. I had a wonderful time working with her.

I am also very grateful to have Professor Ioanis Nikolaidis as my advisor who supported me from the early stages of this project by helping with defining the scope of the project.

A special thanks to My dear Masoud for his continuous encouragement and support. I was very fortunate to work on this research with him.

Finally thanks go out to my loving parents, my beloved sister, my supportive colleagues, and my friends who have prayed for me along the years. Their care and understanding motivated me to continue and finish the work. I truly appreciate it.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We live in an increasingly automated and networked society. Smart environments help this trend to become real by linking computers and other devices to our daily activities and commonplace tasks. Although people have put some efforts on creating smart environments for decades, research on this multidisciplinary topic has become more intense and popular in the last ten years. However, we cannot neglect the tremendous advances in such areas as wireless sensor networking, wireless mobile communications, small and portable devices, robotics, machine learning and human computer interfaces, which have made the dream of smart environments become a reality. A smart environment is a small world embedded with different sensors, which work collaboratively with networked devices to make a comfortable living ambience for the inhabitants. The types of smart environments depend on the purpose and the individuals for which this environment is designed. For example, in some environments the designer may wish to ensure the safety of the inhabitants, while in another the objective may be to reduce the cost of maintaining the environment. One may want to optimize the resource usage, or they may wish to automate the tasks that the inhabitants perform typically in the environment. As this field has attracted many people both in academia and industry, substantial research has been done on different themes of these environments based on market appeal. Some types of smart environments including smart offices [33], classrooms [1], kids room [6] have been designed to permit multiple people to simultaneously collaborate in an interactive experience. Other groups have focused on smart environments to help individuals with health challenges. Some projects in this group are the Gloucester Smart House [2] and Center for Advanced Studies in Adaptive Systems (CASAS) at Washington State University [8]. The Adaptive House at the University of Colorado at Boulder [28, 29] and the Georgia Tech Aware House [23] are grouped in another category of smart homes, which aims to automate activities by learning models of inhabitants obtained from sensors embedded in the house. In this thesis, we focus on “Smart Homes for Assisted Ambient Living”, where the inhabitants are elderly people and the goal is to analyze and visualize sensor data over time and space so that formal and informal caregivers are aware of the occupant’s activities.

1.1 Motivation and Problem

Population aging is progressing rapidly in the world, which has many health consequences such as increasing the dependency of the elderly people [16] and suffering from diseases like Alzheimers and Dementia [12]. Therefore, technologies that can help seniors to live independently have been highlighted these days. With this demographic trend, there is a desire to keep the elderly healthy and living independently in the home of their choice. As the old population grows, the need for resources and facilities designed to take care of older adults will also be increased. Given the cost of nursing home care and the preference of the elderly to remain independent in their homes, it would be cost-effective to consider funds for home modifications. In this work, we are using “Smart Homes” to provide automated assistance for elderly to do their daily activities in their homes. “Smart Home” is a house embedded with several sensors (from motion sensors, door switches, electricity outlets, etc.), which can be used to non-intrusively monitor people’s activities and appliances status. Seniors should be able to complete their key daily activities or ADLs [10], in order to live independently. Daily activities can be categorized into several groups, such as: Personal hygiene, food preparation and consumption, sleeping, medication use, etc. Although, it is very time-consuming to keep track of ADL accomplishment for caregivers, yet it is required as a basic task in an assistive living ambient. Therefore, a system that automatically recognizes ADLs, which allows automatic health monitoring is a necessity. Such a system can be used to help people with dementia by reminding them what they have done, and which steps they should take to complete their activities. Further, it can provide useful information for caregivers regarding the quality and timing of performing ADLs, and alert them when emergencies happen.

In this thesis, we present a visualization system for exploring the sensor data logged from a smart home. The objective is to observe and analyze the activities of elderly people in order to support them in their daily activities, and extract data relevant to their functional assessment and care. To address this issue, our toolkit contains different types of visualizations; from spatio-temporal ones to graphs with statistical properties. One of the challenging questions that might arise in the scope of designing such a visualization system is “what information can be inferred from the visualization product and how?” In our approach to the problem, we will see how some visualization methods can be used to make some interesting inferences and discover amazing patterns among large amount of data.

1.2 Contributions

In this thesis, the main contribution is the development of a web-based framework called “SmartVis” that integrates a variety of visualization components to explore the daily activities of inhabitants in a “Smart Home”. The specific contributions are as follows:

- We develop an activity timeline graph to provide the users with the ability of monitoring the location of each patient and the event triggered by her/him over the time. This graph provides useful information about what each patient is doing in a specific period of time.

- To illustrate how much time the patient is spent on each activity in a day, we present a calendar pie chart. In this view, we present a continuous calendar, where each day contains a pie chart showing the activities performed by the selected resident on that specific day.

- We propose a column chart, which shows the levels of being active for a patient for a specific activity on each day of the week. This chart is also very useful, while the user compares multiple patients in the terms of the time spent on a specific activity.

- We design a heat map graph as one of the visual-analysis tools that addresses privacy issues associated with video surveillance. The graph is generated through observation of the paths traversed by an individual in the specified floorplan over time. To create the graph,we have used a floorplan with the heat-map represented as the intensities of the same color.

- We propose a flexible visualization system that allows users to employ the sensor-logged data as an input from any smart home. Using structured input data sources and XML configuration files has enabled our system to work with both real-time data and archived datasets for individual and multiple residents.

- We finally conduct an extensive simulation-based evaluation of our framework by comparing the results of a performance test (using response time as the metric) on 6 scenarios each contains different loads of data.

In summary, a major contribution of this thesis is providing the user with visualizations for multiple residents along with statistical information on the graphs. To the best of out knowledge, the problem of monitoring multiple people in smart homes is a new issue and has not been covered in the context of smart home visualization systems.

1.3 Outline

The remaining parts of the thesis are organized as follows:

Chapter 2 reviews the earlier work in 3 sections. In the first section, we discuss the specific designs for visualization of sensors both in time and space dimensions. The next part introduces a number of visualization systems for smart homes. And lastly, section 3 focuses on the home medical monitoring systems.

Chapter 3 presents general technical background information, necessary in order to understand the concepts presented in this work. This information covers REST architecture, MQTT messaging and message exchanging using standardized data formats. It also covers protocols and techniques on the web for event-based messaging.

Chapter 4 explains the architecture of the prototype system. It shows our four-tier architecture namely data sources and system configuration layer, activity inference layer, data management layer and analysis and visualization tools. We describe the different components of these layers, their relationships and explain technologies used in each layer.

Chapter 5 describes three typical examples of using our system along with screenshots from various features of SmartVis. In the first part, we show how one can navigate through the system, and visualize graphs for a single resident from the CASAS dataset. The second part illustrates the graphs obtained from monitoring multiple residents in the Smart-Condo™ simulator. And the last section supports the reader with an example of using the Smart-Condo™ real-time data.

Chapter 6 discusses our evaluation using performance test on 6 scenarios with different loads of data.

Chapter 7 is dedicated to concluding remarks and possible future work and extensions. Specifically, we explain how integrating statistical and visualization components into our framework can turn our system to a valuable tool for elderly caregivers in an assisted living environment.

Chapter 2

Related Work

This chapter reviews the related work in three specific areas related to the topic of this research. The first section discusses the research and advancements in the area of spatio-temporal visualization systems including visualization methods. The second section introduces home monitoring systems, a variety of examples in this domain, and the last part describes visualization systems proposed in the area of patient-activity monitoring, along with a comprehensive table that compares different features of these systems and SmartVis.

2.1 Spatio-Temporal Visualization Systems

The most challenging part of our visualization system is to present data both in time and space. This problem has been known for a long time in visualization field, particularly in map-based applications. Regarding this, a large number of visualization systems have been developed. Each of them has their own advantages and limitations based on their goal. Glatsky et al. [15] present a space-time cube considering time as an additional spatial dimension, with a focus on detecting spatio-temporal patterns in event occurrences. In their work, they consider particular type of data, which describes transient events that have spatial and temporal references. The dataset they have used to evaluate their work is a catalogue of earthquakes containing 10550 events. Each data record consists of the time of the earthquake occurrence, the geographic location and a few other thematic attributes. Comparing to their work, our system is working with a larger scale of data, as our dataset covers daily activity events for a person over several years. Also, as they stated, some problems have occurred during the evaluation that indicates their visualization system is not interactive.

Ganti et al [14] analyze the data from a real-world city-scale mobile participatory sensor network comprised of about 2000 taxicabs. Due to long waits for taxis in peak times, they have designed a spatio-temporal heatmap of available taxis, so that people can walk to the closest junction and find an available taxi. In order to keep track of people's movement in a daily life, we have also developed a spatio-temporal heatmap of the space, where the "heat" on a cell location corresponds to the visitation frequency.

In [38] authors present a 2D visualization technique called Storygraph, which provides a view of time and space simultaneously to identify various patterns in data and uncover relationships between different correlating spatio-temporal events. Although this method seems interesting as part of a visualization system, it is not applicable to our system; since it couldn't handle large scales of data.

Sjöbergh and Tanaka [39] describe a system for interactive visualizations of data, which uses pluggable components to easily add new data sources. Twitter is one of the data sources they have used to explain their system. To visualize the data, they have considered a 24 hour clock visualization component, a map and a text query component. In the Twitter dataset, text component shows a specific tweet, whereas map and clock displays the location and time of tweets, respectively. It is very similar to our system as selecting or aggregating data on one visualization component results in propagating data in the other ones automatically. They use XML files for data sources, and this will use a large amount of memory. To address this issue, our system uses a database (ODBC) as a data source along with configuration files for processing the raw data and prepares it for feeding into visualizations components. This will decrease the memory usage significantly. Lastly, their goal in designing the visualizations is different from us, though we find more creativity and aesthetics in their visual components.

Snap-Together [32] provides a user interface, which allows users to choose between various visualizations and coordinates them based on relational joints available in data sources stored in a relational database. Similar to our work, changes made to a component reflect in all the connected components. Since their system provide visualizations for various datasets (website log data, photo library, census data of U.S. states and counties, etc .), the variety of the provided views are limited . For example, for a photo library dataset, they have presented just a scatterplot to view trends on a timeline without focusing on the dataset details.

2.2 Home-Monitoring Systems

As people grow old, they might face physical or social changes that challenge their health. Many of these people don't live with their family, so their loved ones can't visit daily and check their health status. With the increasing number of elderly people and the high cost of nursing care, many industrial and academic groups have been attracted to invest in telehealth systems.

Today, healthcare systems are augmented with more advanced sensors and could track activities in the home. Most research in this area has been done in university testbeds. "Georgia Tech's aware home" [23] contains context-aware applications, where each one fits in one of the two following scenarios: 1) support aging in place, 2) support busy families. These application projects include everyday home assistants, health monitoring and memory aids.

"Welfare Techno House" (WTH) [43] in Japan is another medical smart home project, which attempts to monitor physiological parameters, such as ECG monitoring (without using body surface electrodes) and urine measurement.

"Habitat Intelligent pour la Sante" (HIS) [24] in Grenoble is an apartment which is equipped with both ambient and wearable sensors networked in the smart home to a local area network (LAN). The data obtained from each sensor can be transmitted through a local network to a personal computer. The data is used for the measurement of patient activity to evaluate his/her general health status by considering activity indicators, such as mobility, agitation and displacements.

The “Assisted Living Lab” [31] in Fraunhofer IESE in Kaiserslauter, Germany is a typical apartment for an elderly person. The lab is equipped with a set of sensing, interaction and assistance facilities to provide the base technology and measurement environment to implement and evaluate assisted living scenarios.

The “MavHome” (Managing an Adaptive Versatile Home) project [48] at University of Texas at Arlington focus on conducting research on smart home technologies to consider the environments as an intelligent agent. These environments aim to maintain safety, security and privacy, maximize the comfort of the inhabitants and minimize the consumption of resources.

The “Aging in Place” project at University of Missouri provides health promotion and assistive care services, along with routinely assessing the residents, also coordinating resident’s medicals with physicians and caregivers [26, 36].

The “TigerPlace” [35] is an independent retirement facility in Missouri that is based on the “Aging in Place” model. This smart home is used as a study set for designing user interfaces to display the monitoring data related to the activity level and sleep patterns of older adults [9]. Their application supports a good variety of interactive visual displays including line charts showing activity levels extracted from sensors, pie charts displaying average activity for each area of the house and bubble charts to illustrate bedtime motion,respiration and restlessness over days of the week. In compare to our work, their application covers more details about the sleeping activity, while other daily activities are not distinguished from each other and are presented as a general one.

Another research took place in “TigerPlace”, attempts to evaluate the sensor data displays using 16 heuristic criteria through conducting a user study [3]. They have developed two types of graphs to show the activities of an individual throughout the apartment. The first line graph contains several lines displaying all the motion sensors firings per day for an individual. In the second view, user can select a specific motion sensor and view the line graph for it. The results of the user study shows that flexibility, efficiency of use, navigation and documentation were not well developed in their application, while their visualizations were highly rated for their aesthetic value.

Rantz et al. [34] use the sensor data collected from “TigerPlace” to predict health events such as falls and emergency room visits. Their goal is to generate alerts that notify caregivers of changes in patient’s conditions so they could prevent or delay adverse health events. To achieve the goal, they have designed a sensor network along with a secure web-based interface to display the sensor data for patients, caregivers and researchers. Similar to our system, it allows the user to select a specific patient and a date range. The data can be displayed in different ways including line graphs, pie charts and histograms. Moreover,sensor data are grouped into 4 categories: motion, pulse, breath and restlessness. In our visualizations, we report general condition of a patient in performing daily activities, so our sensor data falls in different categories from this study. Our data grouping is based on normal daily activities such as sleeping, eating, watching TV and so forth. For now, our system does not include a notification and alert component, but we would like to develop it as a future work.

2.3 Patient-Activity Monitoring Visualization Systems

The use of telemedicine over the past decades has been significant. Nowadays, it is very difficult to find a country that has no determined program for developing telemedicine capabilities. Telemedicine is used as a general term to refer to all systems and applications that facilitates communication between patients and caregivers using electronic devices and information exchange to deliver personal health services [4]. These health-enabling systems aim to decrease the cost and improve the health care delivery quality [21]. Companies like IBM, Philips and Intel have been worked actively in the area of remote patient monitoring systems since the early 2000s [5].

Takács et al. [42] propose an ambient facial interface, which provides visual feedback and confirmation to the user in a manner independent of age, culture, language and mental alertness. Their system use animated faces or photographic humans to display emotional facial expressions along with non-verbal feedback, which is easy for an elderly to recognize. These digital faces are controlled by physical measurements or data processed from the current state of the user or the objects and products he or she is interacting with. Using these measurements, finally a single facial expression is displayed to the elderly user, so they can evaluate the quality instantaneously.

Staggers et al. [40] present their work by comparing a text-based user interface and a graphical interface in the health care area. The results reveal that graphical interface significantly improved the error rates, response time and satisfaction rating used in computerized nursing order tasks. Also, it indicates that the rate of learning the system was faster when using the graphical interface.

Producing an AAL system that provides a high-quality of service, requires to consider different aspects of these systems to achieve security, usability, accuracy and interoperability. To identify the essential aspects of AAL systems, a number of surveys of the literatures have been conducted. A recent review by Memon et al. [27] presents a comprehensive literature survey with a focus on health-care frameworks, platforms, standards and quality attributes. They found that many AAL systems are limited to a small set of features; therefore many of the essential aspects of these systems are ignored. In conclusion, they argue that achieving more synergetic AAL solutions requires more user-centred studies, standardized efforts and focus on open systems. In another work by Rashidi et al. [37], a general technical survey on emergence of AAL tools for older adults is proposed. In their paper, they discuss ambient assistive living platforms, algorithms, systems and standards.

Using Wireless Sensor Networks (WSNs) in smart environments provides the opportunity to create pervasive applications, which support the user with scalable and context-aware services. In this domain, Hussain et al. [22] propose a web-based application that uses Received Signal Strength Indicator (RSSI) to investigate localization and mobility. In the experimental study, they consider 9 sensors deployed in a bedroom to determine 1) sleeping behaviour of a person and a limited number of his/her physical activities, 2) occupied chairs in the room and the mobility of the person, and 3) humidity of the room for knowledge extraction. The results are illustrated as line charts, where y-axis shows the RSSI value and time is displayed on the x-axis. As RSSI variation determines the actual behaviour of a person, it is difficult for a naive user to explore information from the line charts without any expert knowledge. Moreover, interactivity among different visualization components is necessary in a web application, which is not considered in their system.

2.3.1 Comparison with Other Ambient Assistive Living Visualizations

In order to compare our framework with other visualization systems, we consider 7 parameters that are usually important in assessing a visualization tool designed for an ambient assistive living. The parameters are defined as follows:

Data Capture / Analysis: The data captured by sensors in the smart home can be processed in two ways based on the user request. In some cases, the user may want to visualize the real-time data for the patients, so the visualizations are generated instantly. Sometimes archived and historical data is required. A good visualization system is the one that can handle both types of data requests.

Interactive Visualization Components: For a visualization system to be considered as interactive, it should meet two criteria. First, the control of some aspects of the visualization components should be available to the user. Second, changes made by the user should be incorporated into the visualization system in a timely manner. This feature is very important in patient-activity visualization systems.

Data Aggregation: In visualization toolkits, a data aggregation component could make the task of data analysis more simple by gathering the information and express it in a summary form for statistical analysis. For example, in this category of data visualization, one type of data aggregation could be reporting the average time spent daily on a specific activity in a range of time.

Multiple Residents Comparison: This feature allows the end user to compare various numbers of patients in terms of activities and daily behaviour in different views. In a rehabilitation hospital, several smart suites are available, where each residency may contain one or more residents. Including this feature in the visualization toolkit would be very advantageous for these kinds of hospitals, as the nurses can investigate if any environment-health issue rises.

Adaptability to Use for Another Smart Home: In designing a software system, it is necessary to develop it in a way, which is adaptable to changes in its environment. An adaptable software system can tolerate changes in its environment without external intervention. In our case, the system should be designed to visualize data from different smart homes, without any need to change the software code.

Automatic Alerts/ Email: One of the components that may seem useful in these types of visualizations is a module that sends automatic alerts or emails, when an emergency occurs. For example, if the sleeping time of the patient exceeds a threshold, the caregiver will receive an alert to check the patient's status.

In this thesis, we compare 6 visualization frameworks with our tool (SmartVis), which are carefully selected to be similar to our system in terms of visualization goals. In this section, first we describe the contribution of each work. Then, we discuss whether each framework considers the aforementioned parameters or not, and finally how they work similar to or different from SmartVis. A brief result of the discussion could be found in Table 2.1.

Gil et al. [17] introduce an enhanced home-based care system by modelling the “busyness” (overall activity) in their place of residence. Based on their research, data mining can play an important role in detecting the changes of the level of daily activities, which directly reflects the changes in health condition. The study found that changes in a busyness metric were visible and detectable even with irregular behaviour. In this regard, they have designed two types of graphs. The first graph is a stacked-column chart showing the “busyness” over different time zones such as sleeping, early morning, late morning, etc . The other graph is a column chart displaying the daily sensor firings for a specific sensor over a couple of weeks, which is very similar to our “activity overview” column chart that shows the daily time spent on an activity over a range of time. The only difference is about the metric used in the SmartVis and their tool.

Mulvenna et al. [30] suggest a data visualization toolkit that supports elderly people with mild dementia in their homes during the hours of darkness. In this work, the main focus is to highlight the fact that different visualization components service different end users requirements. In order to address this fact, they have created a table, which illustrates the issues in communication between AAL services and different user types. In the table, they mentioned that they can capture and analyze both archived and real-time data. For different users, they have considered different types of visualizations that are based on the sensors data. They provide bar charts which shows sensor usage for all the sensors embedded in different areas of the apartment in a daily range of time. Also, they have depicted the daily usage of each sensor. Although, the type of the charts they used to depict the data is similar to ours, SmartVis apply a higher level of data extraction, and provide more information rather than raw data obtained from the sensor readings. SmartVis is capable of showing the triggered events and moreover recognize activities performed from a sequence of events. They have designed an alarm dashboard for different group of users for emergency incidents or unexpected behaviours.

The activPAL™[1] is a graphical visualization system that represents daily activities of a person categorized into 3 groups of sitting, standing and walking. This information can be used to track any changes in the free-living activities against medication or treatment regimens. The technology records and identifies periods of moderate physical activity throughout a day, and also provides cumulative totals of these periods. Activity for each hour is represented by bars covering 15 second periods, and the pie chart visually illustrates the total time in an hour spent on each primary activity. Their system is similar to us in answering questions based on user's daily or weekly profiles, for example, has the user changes his/her level of activity compared to the previous days? But, it couldn't cover detailed activities such as watching TV, meal preparation, taking shower etc . Moreover, the visualization graphs seem like an activity record, and no interactivity is mentioned in the documents.

Tunstall's ADLife[2] is an ADL monitoring system, which unobtrusively monitors room transitions, door usage, electrical appliance usage and bed/chair occupancy. The user can view the data obtained from the sensors in different ways including simple traffic light table , which displays the sensor usage for different range of times, or bar charts showing all sensor activities for a specific day. This system is also enabled to send automatic email alerts if the user's ADL patterns deviates from the normal levels. Similar to [30], this system cannot display the activities percieved from the events.

More relevant to our application, is the work of Thomas et al. [44] on presenting a visualization tool for smart home called PyVis. This visualizer provides an interactive interface to visualize smart home in real-time as well as providing historical trends. Our approach is similar in spirit, but we pay more attention to the activities performed by each resident in the smart home, while PyVis only keeps track of environmental changes.

In the paper proposed by Suryadevara et al. [41], a mechanism for esti-

[1] http://www.paltech.plus.com/products.htm

[2] http://www.tunstall.co.uk/Uploads/Documents/ADLife%20solution%20sheet%2002.01.13.pdf

Table 2.1: Comparison of patient-activity monitoring visualization systems, including our system (SmartVis).

Frame Work	[30]	[44]	SmartVis	activPAL	ADLife	[17]	[41]
Data Capture/ Analysis	Real-time/ Archive	Real-time/ Archive	Real-time/ Archive	Archive	Archive	Archive	Real-time/ Archive
Interactive Visualization Components	Yes	Yes	Yes	No	Yes	No	Yes
Data Aggregation	No	Yes	Yes	Yes	No	No	Yes
Multiple Residents Comparison	No	No	Yes	No	No	No	No
Adaptability to Use for Another Smart Home	N/A	Yes	Yes	N/A	N/A	Yes	Yes
Automatic Alerts / Email	Yes	No	No	No	Yes	No	No

mation of elderly well-being condition is reported, which is based on usage of house-hold appliances connected through various sensing units. In this research, they have defined "wellness" as well-being of elderly in performing their daily activities efficiently at home, and to measure it, they have defined two functions. The first function measures the "wellness" of an older adult based on the inactive duration of the appliances; while the other function works according to the excess usage measurement of appliances. Also, they have developed an interface, where the real-time activity status of the elderly can be seen on. This visualization system stores the sensor activity information and analyzes the "wellness" indices, as well as remotely monitoring the elderly activities. Similar to our system, they have developed a colorful bar graph displaying activity occurences for different days over 24 hours. They have measured the "wellness" of different patients at different smart homes. Moreover, they have prepared a line chart measuring the excess usage of appliances for 7 days of the week while the patient is working with different house-hold appliances.

Chapter 3

Technology Background

This chapter reviews the technologies we have used in developing our system. The first section describes the REST architecture, and it's evolution through the time. The next section introduces two popular data formats, namely XML and JSON. Finally, the last section discusses the web messaging protocols, specifically the MQTT protocol.

3.1 REST Architecture

New technologies like Wireless sensor networks, RFID and real-time localization are becoming more common and popular these days. The proliferation of these devices in a communicating-actuating network creates the Internet of Things (IoT), wherein sensors and actuators are seamlessly integrated into the information network [19]. The IoT enables smart environments to identify and recognize objects, and retrieve information from the Internet to facilitate their adaptive functionality [47]. The Web of Things (WoT) [20] is a notion that allows real-world objects to be part of the World Wide Web. It provides an application layer that facilitates the creation of Internet of Things applications. The WoT mainly focus on four levels. First, it attempts to connect embedded devices to the internet. In the second step, it embeds web servers on the mentioned devices. Third, it models the offered services by these devices in a resource-oriented way. And finally, it exposes these services as RESTful web resources.

Web services have been vastly used in the recent years to integrate distributed system. They can be categorized into two major classes:

– SOAP-based services [11], in which the service may expose a set of complex standards and specifications for enterprise application integration.

– REST web services, which is an architectural style explaining how to use HTTP properly as an application protocol.

REST suggests the use of a uniform interface. Resources can only be manipulated by the methods defined in the HTTP standard. The four most important methods are:

– GET is used to request a representation of the specified resource

– POST represents an update or insert of a resource.

– PUT is useful to alter the state of a resource.

– DELETE is used to delete the resources.

The simplicity, the use of a uniform interface and the wide availability of HTTP libraries, have made it loosely coupled and reusable.

3.2 Exchange Data Formats on the Web

Exchange of data between two or applications on the web should be in a machine-readable format. Many data formats have been proposed, where each one aims in optimizing the speed or generality. In the following subsections, we describe the most popular data formats, which are also employed in our software (see section 4.1).

3.2.1 XML

Extensible Markup Language (XML) is used to describe data. XML is a markup language that defines a set of rules for encoding documents in a format, which is readable for both human and machine. XML code is a formal recommendation from the World Wide Web Consortium (W3C) [7]. XML data is known as self-describing, which means that the structure of the data is embedded within the data, so when you want to add a new data, there is no need to rebuild the structure to store the data. The basic building block of an XML document is an element, which is enclosed by tags. Element names describe the content of the element, and the structure indicates the relationship between the elements. Listing A.2 shows an example XML listing all the individual sensors embedded in the Smart-Condo™ . We can almost show any kind of data in XML, if we use XSL along with, which is a language for expressing style sheets, and describes how to display an XML document of a given type.

3.2.2 JSON

JavaScript Object Notation (JSON) [18] is a human-readable standard data format that transmits data objects such as associative arrays. Unlike XML, JSON tries to remove the use of tags, and minimize the amount of the characters used for the serialization of a data object. A JSON object is usually a key/value pair, which is a chunk of data stored along with a look up key. Listing 3.1 shows a JSON object (Polygons) holding a list of two JSON objects (Polygon). This is part of our code used to generate heat-map floorplans. Each floorplan is consisted of polygons showing walls and obstacles available in the apartment. Walls are represented as orange polygons, while obstacles such as sofa, TV and etc. are considered as grey ones (Figure 5.14). Moreover, to draw each polygon, vertex coordinates are required, which is expressed as “points" in our JSON file.

Listing 3.1: Example JSON object with a list (Polygons) of two JSON objects (Polygon)

```
{
"Polygons": [{
  "name": "polygon 1",
  "fill": "orange",
    "points": [{"x": "0", "y": "3.23"},
      {"x": "4.7", "y": "3.23"},
      {"x": "4.7", "y": "2.99"},
      {"x": "4.57", "y": "2.99"},
      {"x": "4.57", "y": "3.13"},
```

```
      {"x": "0", "y": "3.13"},
      {"x": "0", "y": "3.23"}]
 },
 {
  "name": "polygon 2",
  "fill": "orange",
    "points": [{"x": "4.57", "y": "2.19"},
      {"x": "4.7", "y": "2.19"},
      {"x": "4.7", "y": "2"},
      {"x": "4.98", "y": "2"},
      {"x": "4.98", "y": "1.9"},
      {"x": "4.57", "y": "1.9"},
      {"x": "4.57", "y": "2.19"}]
}]
}
```

Simplicity, efficiency and readability of JSON have made it an extensively used data format for the transmission of character data. Finally, to define the structure of JSON data for validation, documentation, and interaction control, JSON Schema is a specification for a JSON-based format.

3.3 Web Messaging Protocols

The HTTP protocol acts as a request-response protocol in a client-server model. In this model, the initial request for data begins with the client, and then is responded by the server. This style of network communication is called client pull technique. However, HTTP doesn't seem suitable for applications with event-driven scenarios. In such cases, a push technology seems more useful. Server push [13] is a style of Internet-based communication, where the request for a transaction is initiated by the publisher.

MQTT Protocol: MQTT (Message Queue Telemetry Transport) [45] was originally developed out of IBM's pervasive computing team and their cooperation with their partners in the industrial sectors. In the recent years, the protocol has been presented as an open source messaging protocol, and is moving towards becoming a standard protocol.

The MQTT is a publish/subscribe protocol specifically designed for resource-constrained devices and low bandwidth, high latency or unreliable networks, which makes it effective in embedded systems. MQTT is a useful messaging protocol in home automation and small device scenarios. It can be used in sensors communicating to a broker via satellite link, and also over occasional dial-up connections with healthcare providers. It is also ideal for mobile applications because of its small size, low power usage, minimized data packets, and efficient distribution of information to one or many receivers.

Chapter 4

Software Architecture

We have developed an application prototype that can be used as a tool for performing different statistical analysis and visualization tasks on ambient-assistive living traces. This chapter describes the architecture of the system. Figure 4.1 illustrates the four-layer architecture of our web-based application, including data sources and system configuration layer, activity-location inference layer, data management layer and analysis and visualization tools. Each of these layers and their functionalities/components are explained in subsequent sections. We also briefly explain the technologies we have used in different layers.

4.1 Data Sources and System Configuration Layer

The first layer corresponds to wireless sensor networks (in case of an actual deployment) or sensor events generator (in case of using a simulator) or sample data (in case of using an external database of events).

Wireless Sensor Network: In this layer, we have a wireless sensor network, where on nodes, sensors can be attached. Each node can be equipped with digital or analog sensors, and can communicate with a central gateway attached to a server, where all data can be logged on. When an event is sent, it means that a digital sensor value has changed its state, or a threshold on an analog sensor has been violated.

The sensors in the wireless sensor network have a star topology, therefore

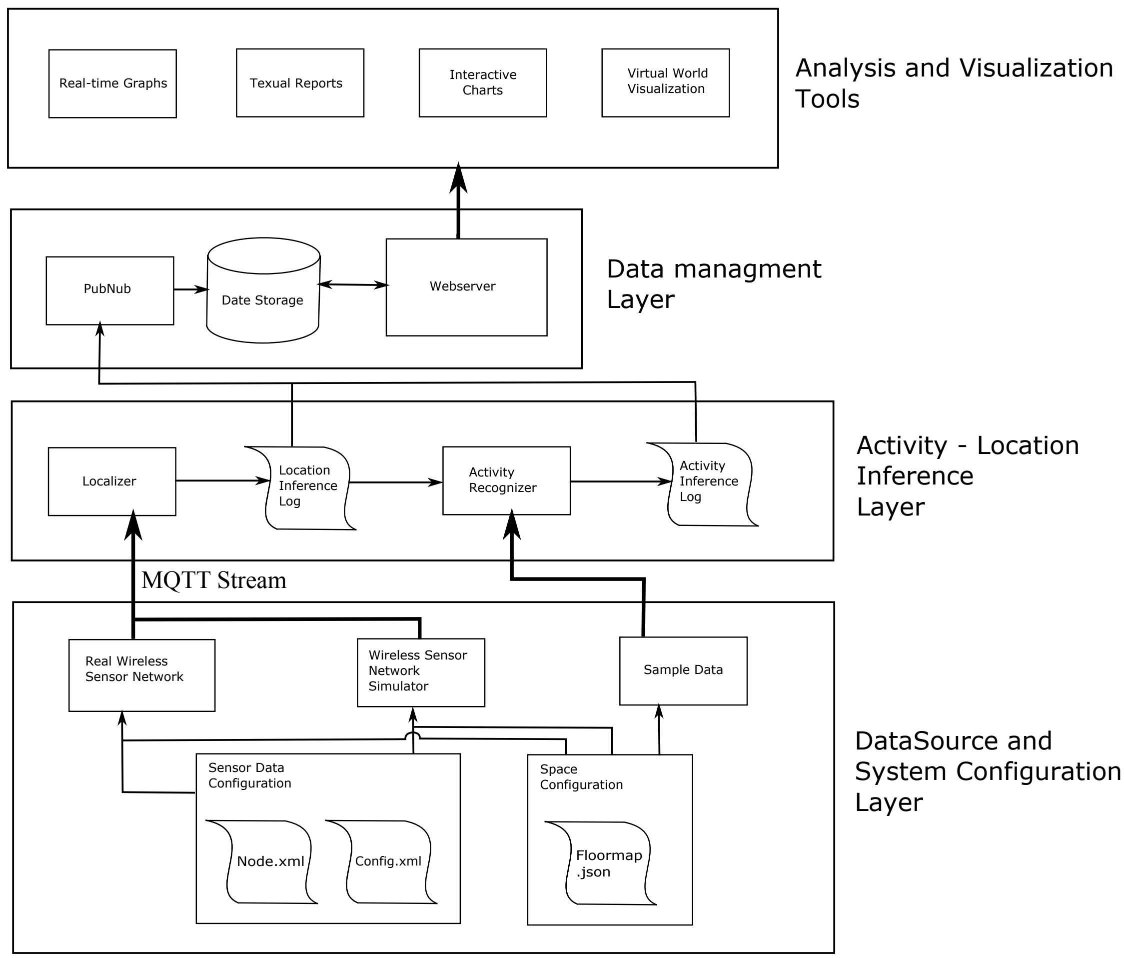

Figure 4.1: System Architecture

all the collected data from these sensors is transmitted to the sink node, which is connected to the component bridge [46].

Wireless Sensor Network Simulator: Simulation framework enables us to generate sensor data without having an actual deployment. In order to have a large dataset of sensor readings, we need participation of human, which is difficult to organize. Thus, a simulation-based alternative gives us the opportunity to conduct arbitrary experiments without having a real deployment. The simulator is capable of generating sensor placement based on a 2D model of space. For a more detailed description of the simulation software used in our platform please refer to the thesis by Vlasenko [46].

Sample Data: In addition to the real sensor network and the simulator, our system is capable of working with other sensor-logged datasets collected from smart homes. These datasets should at least contain 4 fields of data, which are: 1) resident ID, 2) apartment name or number, 3) type of the sensor triggered, and 4) date and time of the event. Along with the dataset, the sensor type configuration files and the floorplan of the apartment are also required. We use the primary dataset and the mentioned files to extract the system actual input data through a preprocessing step. To evaluate our framework with this category of data, we have selected a dataset from WSU CASAS [8]. Specifically, the dataset has been recorded in the Kyoto testbed, and includes sensor events collected from WSU apartment for two residents living there during the 2009-2010 academic year. More detailed specification of the dataset can be found in Chapter 6.

Sensor Data Configuration: In our system, we consider two types of configuration files that provide extra information for the real network and the simulator modules. The first file (Config.xml) contains basic information about the sensor ids, sensor types and their corresponding hardware characteristics. The second file (Node.xml) provides location of each sensor installed on a node, and also describes on/off commands for both the real and the virtual world. Moreover, it keeps the size, id, and the area of the installed sensor (See Appendix A).

The main purpose of using these files is to reduce the cost of implementation, and make a flexible toolkit that works with various sensory infrastructures.

Space Configuration: In order to generate a floorplan for the heat-map visualization, we need to have the exact coordinates of the walls and obstacles in the apartment. For this purpose, we consider a JSON file similar to Figure 3.1 containing walls and obstacles information as “polygon” with 2 features: 1) “fill”, representing the color of the “polygon” in the floormap, 2) “points”, showing the vertex coordinates.

MQTT Stream: MQTT protocol supports publish/subscribe pattern, which is an alternative to the traditional client-server model. In this paradigm, publisher and subscriber are not aware of the existence of each other, so there is a third component called broker, which is known by both the publisher and the subscriber, and acts as a filter on the incoming messages and distributes them accordingly. The filtering is based on a topic, which is part of the message. The publishing client updates under certain topics, and the receiving client subscribes on the specific topics, which is interested in to get the updates. In our system, first the real sensor network or the simulator publishes sensor readings under a set of topics, which are defined as “node id”s in Listing A.2. Then,through an API, which has previously subscribed to those topics, the updated events will transfer into the activity-location inference layer, particularly into the localizer component. A sample of a sensor reading published as an MQTT message is as follows: “Message: 62,16,1441148623000,0”. The first number in the message shows the topic number, the second indicates the sensor type, the third displays the timestamp and the forth one is the status of the sensor. If you refer to Listing A.2, you can find a sensor with “id” = 62 and “sensor” = 16, which is an RFID reader. The whole sample message states that the RFID reader tagged by “id=62” has become off on Tuesday , 01 Sep 2015 23:03:43 GMT.

4.2 Activity-Location inference Layer

Localizer: The localizer component receives the raw sensor readings from the MQTT stream, and calculates location estimates for each person. If the readings are generated from a switch or pressure sensor, then it's location is the actual position of the sensor event. Readings from the motion sensors and RFIDs will be concatenated to form a binary string. Each motion sensor/RFID reading is a binary value, which indicates its on/off status. The initial estimate of localization algorithm is the center of mass of the polygons, which the recent triggered sensors/RFIDs cover. In the estimation mechanism, a constant-sized window is considered, which contains a collection of previous locations.

When the window is full, the binary string is processed to estimate the new coordinates of people's location. This process is repeated until reaching the next adjacent triggered area.

The information received from the sensor readings and localizer component is stored in the "Location Inference Log". Particularly, this log is responsible to answer the following questions when a sensor event occurred:

- Who has triggered the sensors?
- What time and date has the event started and finished?
- What was the event?
- In which area of the apartment has this event triggered?
- What was the exact coordinate of the performer during this event?

Activity Recognizer: The Activity Recognizer component receives location inference information, and generates activities based on the events. Here, an activity is defined as a set of events that have triggered by a single person consecutively. These activities are recorded in an "Activity Inference Log" to be stored in the database. The generated datasets by the Smart-Condo™ simulator or the Smart-Condo™ real-time doesn't provide the performed activities. These datasets contain a sequence of events, which requires to be processed to obtain the activity. In order to extract the activities, we categorize the activities into 2 groups: simple and complicated activities. For example, an activity like “sleep” is defined as two smaller events: 1) Lie down on the bed, 2) Get up from the bed, and is counted as a simple activity. However, some activities need more details to be considered. “Meal Preparation” is a complicated activity, since different scenarios can make this activity happen. One scenario happens when the resident is making a salad. For this case, the sequence of events would be: “Open the fridge door”, “Move in the kitchen”, “close the fridge door”, “Move in the kitchen”. In the other scenario, the resident is going to make a meal. In this case, the sequence of events seems more complicated, as the user needs to use the stove, too. Besides, another complexity rises when the user triggered an event related to an action, but the time spent on doing that event is not sufficient to consider it as a part of activity. This issue makes us to design our activity recognition component with time thresholds.

In our proposed algorithm, for each action A_i, we consider an ordered set of events $E_i = e_1, e_2, e_3, \ldots e_m$, which usually occurs in performing an activity. As we have a limited number of activities and sensor events, generating these sets are done manually. In the next step, we assign a minimum time threshold for a pair of priority event. Priority events are those events that are not triggered by a motion sensor. For instance, “Sit on the Sofa” and “ Get up from the Sofa” is a priority event pair, and we have assigned a minimum time threshold of 10 seconds. Lastly, to recognize an activity, if the time spent on a priority pair event is bigger than the minimum threshold, then we label that sequence of events by A_i; otherwise we will label it as a “walking” action.

4.3 Data Management Layer

Data Storage: The logged data from the previous level needs to be kept in a relational database such as MySQL. MySQL is a relational database that we have used in our system to record sensor readings, as well as information on the extracted activities. In the database, we store two tables for each dataset. The table design allows easy inserting or deleting users and activities without affecting any other tables. The first table is the “Sensor Events” table, which is responsible to store the sensor readings received from the MQTT messages. When an event occurs, the name of the action performer (e.g. R1), the name of the event triggered by the sensors (e.g. Bathtub on), the area and the exact coordinates of the place (e.g. Bathroom, x: 6.67, y: 1.87), and also the date and time of the event is stored in the “Sensor Events” table. The second table is the “Activity” table, containing the name of the action (e.g. Meal Preparation), the performer, the area, and the start and end time and date of the action. Here, in this thesis, we address the sensor reading by “event”, and a sequence of events creates an “action”.

PubNub [1]**:** In order to create real-time visualizations, we have used PubNub, which is a real-time network. PubNub utilizes a Publish/Subscribe model for real-time data streaming, which includes 3 atomic components namely API keys, messages and channels. If a client needs to subscribe and not publish, then it is sufficient for the client to initialize with the“*publish_key*”. For those clients who will be publishing, or publishing and subscribing, it will need to initialize with both “*publish_key*” and “*subscribe_key*”. When you register an application in your PubNub account, a “*publish_key*” and “*subscribe_key*” is automatically assigned to your application. After initializing API keys, a publishing client publish messages to a given channel, and a subscribing client will only receive those messages associated with the channels it has subscribed to. In our system, the localizer component publishes the messages and our visualization module receives them through an API (See B.1 for more details).

[1] http://www.pubnub.com/

4.4 Analysis and Visualization Tools

4.4.1 Virtual World Visualization

One of the visualization- analysis tools that have been used in the Smart-Condo™ platform is the virtual world animation. The idea behind this type of visualization is to keep user's privacy, while providing highly detailed animations of user activities. This visualization tool uses an API to get the parsed sensor readings from the database, converts it to readable commands and sends them to the virtual world objects and avatars.

4.4.2 Textual Reports

Textual reports provide information on patient's activities with specific date, time, location and also the type of the activity that has been performed. These reports include a precise and detailed list of patient's activities that helps caregivers to check the patients health condition regularly.

4.4.3 Interactive Charts

The main focus of this thesis is on developing interactive charts that can analyze and visualize patient daily activities. In our work, we have integrated various charts with different purposes for both real-time and archived data. Through these graphs, caregivers can keep track of 1) the patient's location in the apartment (heatmap graph), 2) the time spent on each activity per day (calendar-pie chart), 3) accurate time and location of the triggered event when the patient is moving or performing an activity (line chart), and finally 4) the changes on the time spent on a specific activity during a week/month (column chart). More details on these charts are provided in chapter 5.

Goals

The overall purpose of designing this system is to construct a flexible tool to enhance ease of access to powerful analysis of patient's activities from a spatio-temporal perspective. Consequently, it considers several aspects that are important for analyzing daily activities of a person in the context of health monitoring. By using different display techniques, the visualization system shows when, how often, in what contexts and where individuals perform activities in various time granularities.

Our method is flexible: First, it allows the user to select a range of individuals from a single patient up to the whole patients available in the residence. This option enables nurses to have the advantage of monitoring multiple patients simultaneously. Second, it is possible to illustrate activities on different levels of detail. In our case, 13 actions are considered for the patient performing in the apartment, which are: 1-Sleeping on the bed, 2-Resting on the sofa, 3-Bathing, 4-Personal Hygiene, 5-Meal preparation, 6-Eating, 7-Walking in the room, 8-Working, 9-Entering the apartment, 10-Leaving the apartment, 11-Housekeeping, 12-Watching TV and 13-Walking from the bed to toilet. Sometimes, the user doesn't want to see the activities in this detail, thus they can be categorized as more general activities into a smaller number of groups. Specifically, we have sectioned the above activities into 3 groups: 1-Active, 2-Sleep, 3-Out , for those users who needs a high level of activity information. Third, according to the motion sensors data, it is possible to detect what rooms in the apartment patients do visit during performing their activities. Furthermore, frequency of visiting a place is also considered in our visualizations.

Forth, it permits the user to visualize activities over different time granularities. Moreover, user can select an arbitrary range of days. This option helps to display the required information and avoids messy charts.

Data Representation

To develop and implement a visualization system, a primary sketch is required, which is usually adopted by the user requirements. End users requirements are the basic skeleton of any software. Here, we collected a number of questions that the end users are interested in:

– Q: How did the patient spend his time? How much time has he devoted to perform each activity?

– A: To answer this question, we have developed a pie chart, which can show the time patient devoted to perform each activity. Activities are separated by different colors, and by moving the cursor over each portion of the pie, name of the activity and the spent time will be displayed. This chart gives complete information about the living habit of the patient.

– Q: Does the new condition (e.g. taking new pills) have any influences on the patient's activities? If so, how frequent was it?

– A: A column chart, where time is on the x-axis and the spent time on doing the specific activity is on the y-axis, should answer this one. Through this graph, the user can visualize how the patient changes his activity durations due to a specific reason.

– Q: When does the patient do a specific activity?

– A: Sometimes, the nurses or the patient's family like to know when the patient does an activity. We will illustrate this information by a line chart, where x-axis shows the time and y-axis displays different activities. Based on this chart, nurses know when the patient has taken his pills or does he immediately sleep after having lunch.

– Q: Where does the patient usually spend his time?

– A: In response to this question, we have developed a heatmap chart. This chart displays how the patient is moving through different rooms in the apartment. A colored square in the apartment map means the patient has passed or spent some time there. The darker the color is, the more frequent that point is visited.

System Functionalities

In our toolkit, we consider several functionalities, which we are listed as follows:

Dataset selection: Our framework is flexible to work with several datasets. Users can select among the available datasets through a dropdown list. Also, they can choose between individual or comparison modes of display. In individual mode, users can only visualize the graphs and reports for a single resident, whereas in comparison mode, different residents can be compared together in different graph layouts.

Display multiple datasets: In this toolkit, users can visualize graphs and analyses data from various datasets in different tabs. For example, one tab may contain graphs for a single resident of CASAS dataset, while the other tab compares the charts and reports from the Smart-Condo™ data for 2 individuals. This functionality makes our system user friendly, as the user is able to visualize multiple projects results just in one window.

Display Data as a Graph: For developing this functionality, we use High-Charts [2] and D3js [3] libraries to draw a graph based on the retrieved information from the database. To draw each graph, several queries are designed in the PHP scripts to get the desired data over an Ajax request and render it in the appropriate graph.

User Interactions: User interaction is an important aspect of data visualization. Visualizations especially for exploratory purposes, is usually inefficient without interactions. It is not sufficient to only visually display the data; the user needs to interactively explore the data. The ability to select and manipulate subsets of data, and change viewing parameters interactively supports the user in achieving insight into the details and characteristics of the data. User interactions can be simple as setting display features or changing view parameters, or it may be complicated as navigating through multiple views.

[2] http://www.highcharts.com/
[3] http://d3js.org/

Zooming views: In visualization, zooming is a powerful and essential function for users to navigate and explore the data. In our tool, we implement a zooming function for the “Timeline Activity” chart, which facilitates the navigation of the user through more detailed data.

Chapter 5

Three Typical System-User Interaction Scenarios

In this chapter, we present three demonstration scenarios to illustrate how SmartVis works. As our toolkit supports views for both individual and multiple residents, we devote one scenario to each mode. Moreover, we prepare screenshots of graphs working with real-time data. The order of the scenarios is as follows: First, we show how we can navigate through the system and explore the visualizations for an individual using the CASAS dataset. Second, graphs and reports generated by monitoring multiple residents in the Smart-Condo™ are discussed. Finally, we explain the views provided to use with real-time data followed by screenshots of the graphs. Figure 5.1 shows a view of our dashboard for the Smart-Condo™ Simulator dataset comparing 2 residents.

5.1 Scenario 1:Single Resident (External Data)

In this section, we prepare a scenario to show the required steps for monitoring a single resident using the CASAS dataset.Unfortunately, for this dataset, we didn't have access to the resident's location coordinates while moving in the apartment; therefore, for this scenario, heat-map graph is not provided. In the next part, we describe how to set the system configurations and analyze the graphs by providing more details.

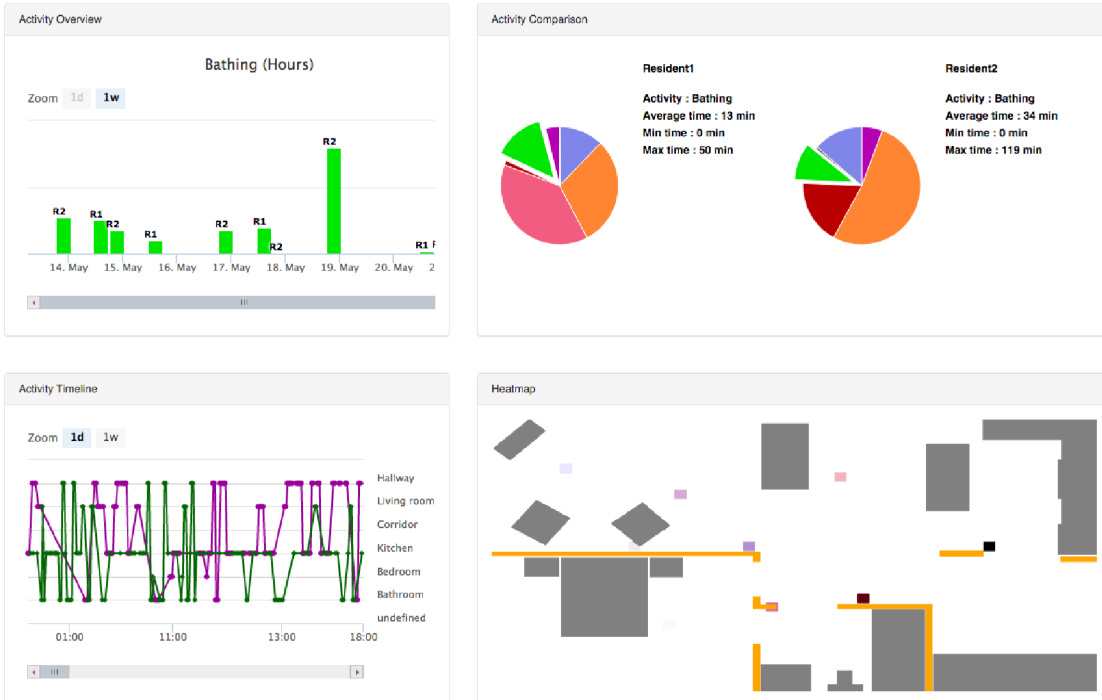

Figure 5.1: A view of the dashboard for the Smart-Condo™ Simulator dataset in comparison mode

5.1.1 View Setup

First, the user opens the web browser and loads the dashboard. In the next step, the dataset (Figure 5.2) and the view mode (Figure 5.3) should be selected. The selected data contains a number of time-stamped events that are labeled with corresponding locations and residents. To visualize data, the user should choose an arbitrary apartment (Figure 5.4) and resident (Figure 5.5), a time period and a special activity (Figure 5.6) that wants to get more information about. When the user clicks on the "Draw Graphs" button, depending on the selected parameters, visualizations are loaded and displayed in their corresponding panels. For instance, "Activity timeline" graph is depicted in Figure 5.9.

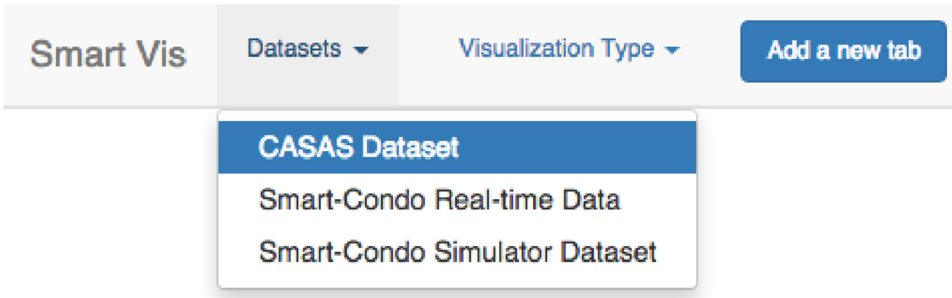

Figure 5.2: Dataset Selection Menu

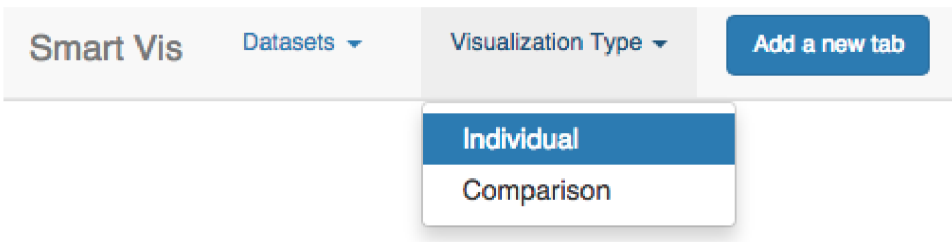

Figure 5.3: View Mode Menu

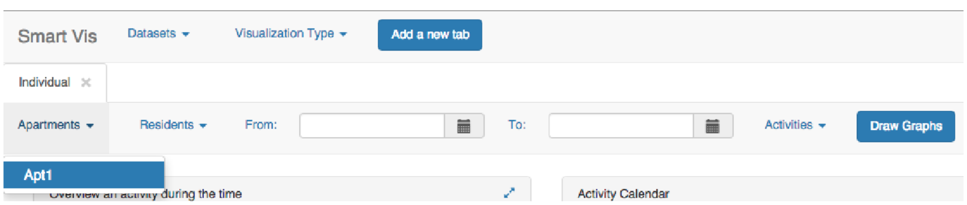

Figure 5.4: Apartment Selection

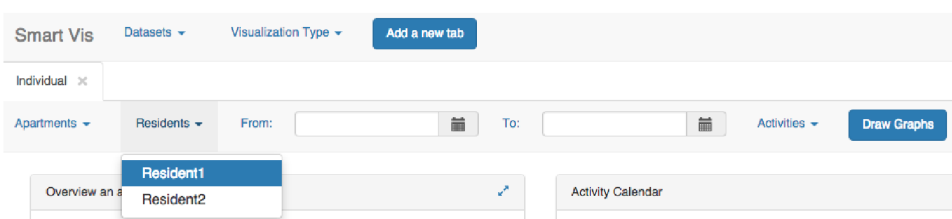

Figure 5.5: Resident Selection

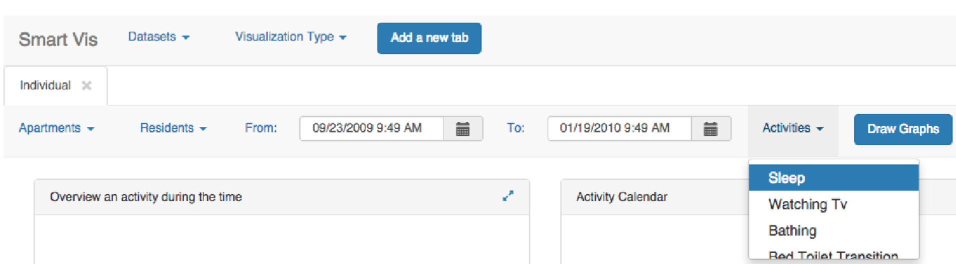

Figure 5.6: Time and Activity Selection

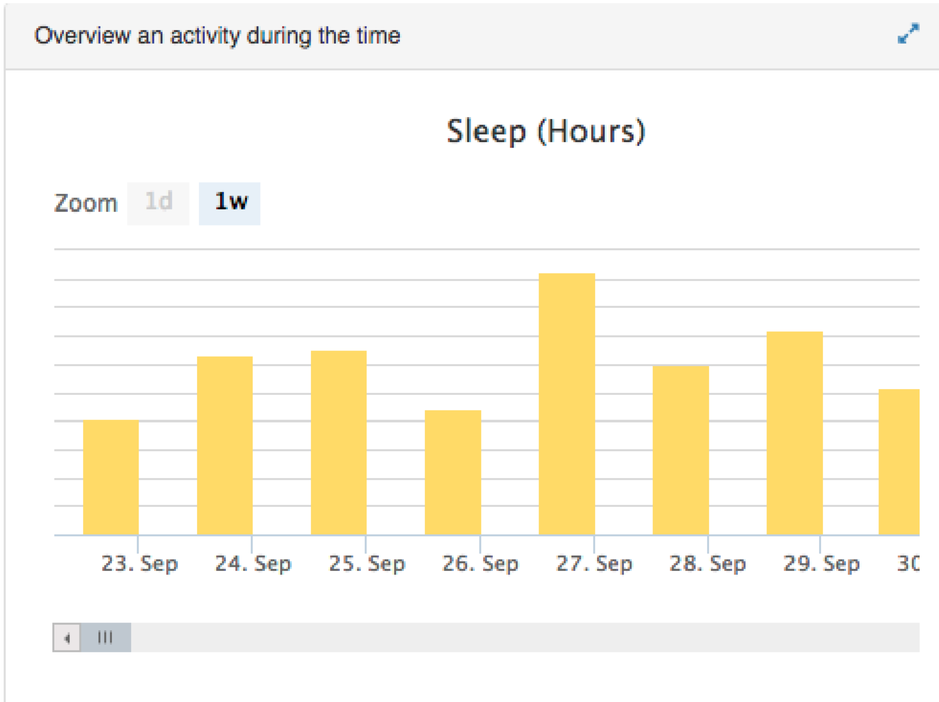

Figure 5.7: Activity overview for a single resident during a week

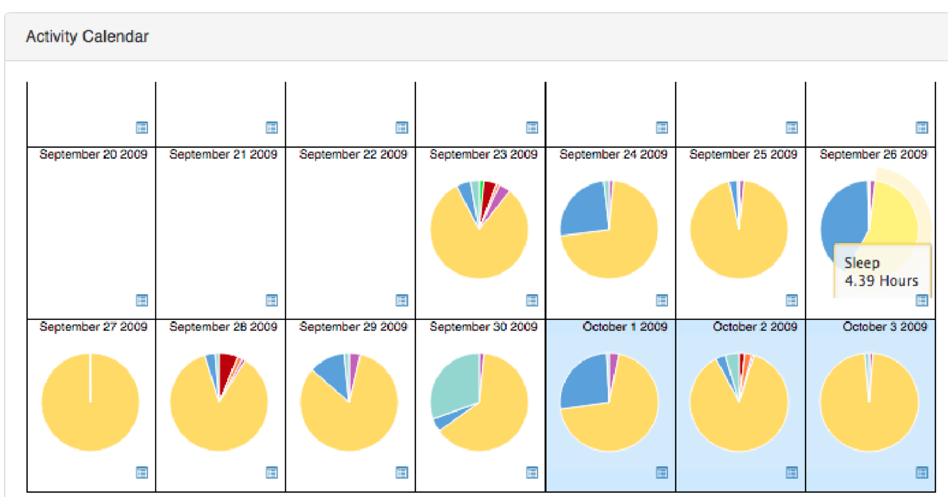

Figure 5.8: Activity pie chart for a single resident

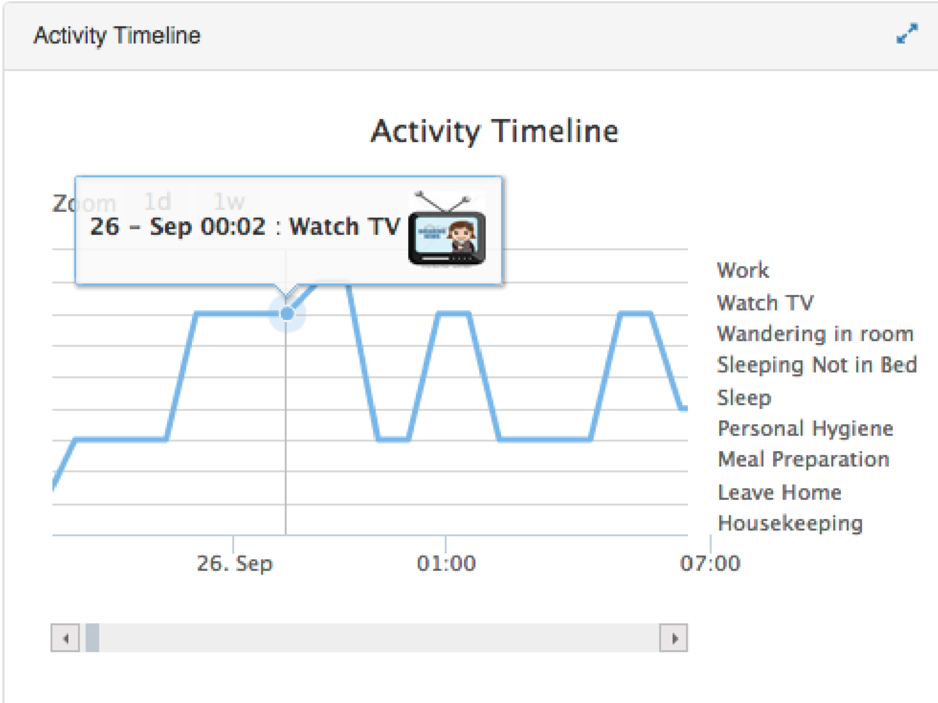

Figure 5.9: Activity timeline graph for a single resident

5.1.2 Graphs and Views

Each visualization graph offers different functionalities. In the activity calendar graph (Figure 5.8), when the user right-clicks on a cell of the calendar, which represents a day, the activity-timeline and the heat-map graph will send a new query to database and display the new visualizations for that specific date. Also, the clicked date on the calendar will be highlighted. By left clicking on a cell, a notepad is opened, where the user can add, modify or delete his/her notes about the patient on that day. For each patient, a note folder exists containing all the notes saved under his/her name. A view of notepad can be found in Figure 5.10.

On the header panel of the “Activity Timeline” graph, a list icon is placed. The purpose of designing this icon is to present a complete report of the patient’s activities during a day with time-stamped details, when clicked (Figure 5.11).

To make a harmony among all the visualizations, we consider a unique color

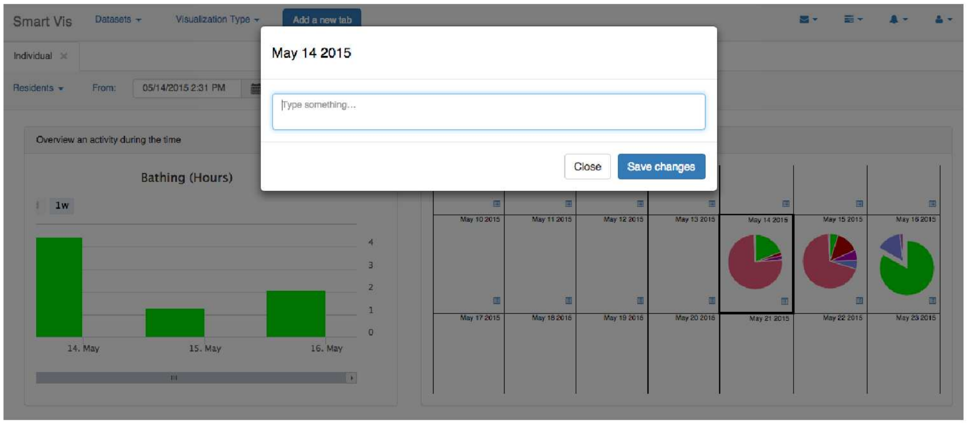

Figure 5.10: A view of the daily notepad

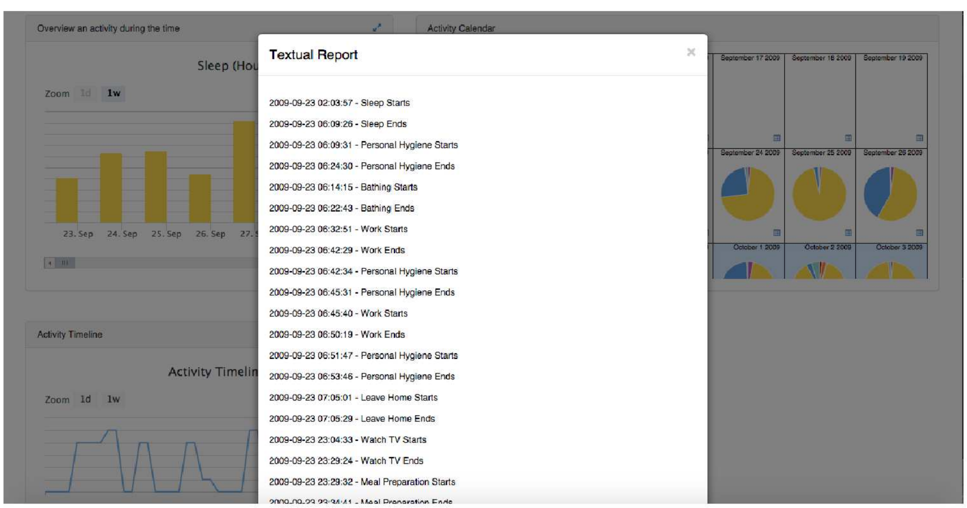

Figure 5.11: A view of the textual report

for each activity. In fact, SmartVis is capable of assigning a unique color for up to 30 different activities through SQL queries. The user can select an activity from the selection panel, and review the spent time on the selected activity over a chosen period of time in the “overview activity” column chart (Figure 5.7). Now, if the user wants to select another activity, there is no need to go to the selection panel. In the activity calendar, each portion of the pie represents an activity (Figure 5.8), where by clicking on it, a new activity is chosen and the changes reflects in the overview activity graph.

In this toolkit, we aim to provide all the graphs in a non-sliding web page. This feature supports the user with a whole picture of our system without bothering him/her to scroll over multiple pages. By applying this feature to our work, our graphs may look small and seems hard for the user to follow the details. To overcome this difficulty, we have embedded an expansion icon in the header panel of each graph, where the user can find a larger view of the graph by clicking on it.

5.2 Scenario 2: Multiple Residents (Smart-Condo™ Simulator Data)

This scenario shows the capability of our system in comparing and analyzing multiple people’s activities. As far as we know, we have developed the first patient-activity monitoring visualization system that supports multiple residents.

5.2.1 View Setup

In this scenario, after loading the dashboard, the user selects Smart-Condo™ Simulator dataset and the comparison mode. When the new tab appears, the user can select 2-4 arbitrary residents from the checkbox drop down list. The selected residents can be from the same residency or live in different suites. Like the previous scenario, the visualizations will be loaded after selecting the time period and an arbitrary activity.

5.2.2 Graphs and Views

As we mentioned before, the Smart-Condo™ is embedded with various sensors, so that we can keep track of different events triggered by a resident. Using these sensors would be sufficient, if only one patient resided in the Condo. For two or more residents that use a common suite to live, we need RFIDs to detect different residents. Radio-frequency identification (RFID) is the wireless use of electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. In our case, each patient is wearing a tag, which distinguishes him/her from the others. As there are a limited number of RFIDs embedded in the Condo, there is a possibility of missing the identity of the activity performer in some events. Therefore, a component is developed in the Smart-Condo™ simulator, which tries to remove any ambiguities, while the patients are moving in the Condo.

The graphs and visualizations designed for this scenario have more complex features due to the ambiguity. In the"activity timeline" graph, each line represents a patient, which is presented in Figure 5.12 for 2 different residents. To compare the time each patient spent on performing different activities, a pie-chat is designed. If the user selects a period of time (more than one day), then along with this graph, a statistical report will be generated showing the average, minimum and maximum time the patient spent on a activity. Figure 5.13 displays the mentioned feature for 2 residents.

The most interesting visualization graph is the heatmap, which shows each person by a different color. As we discussed before, the color intensity of a point determines the frequency of visiting that point. Now, if we have more than one patient, and a point in the Condo is visited by more than one person, then the color of the point is the average color of heat-maps for each patient passed on that point. Figure 5.14 shows the heatmap graph for the selected 2 residents.

The last visualization in this scenario is the column charts comparing time

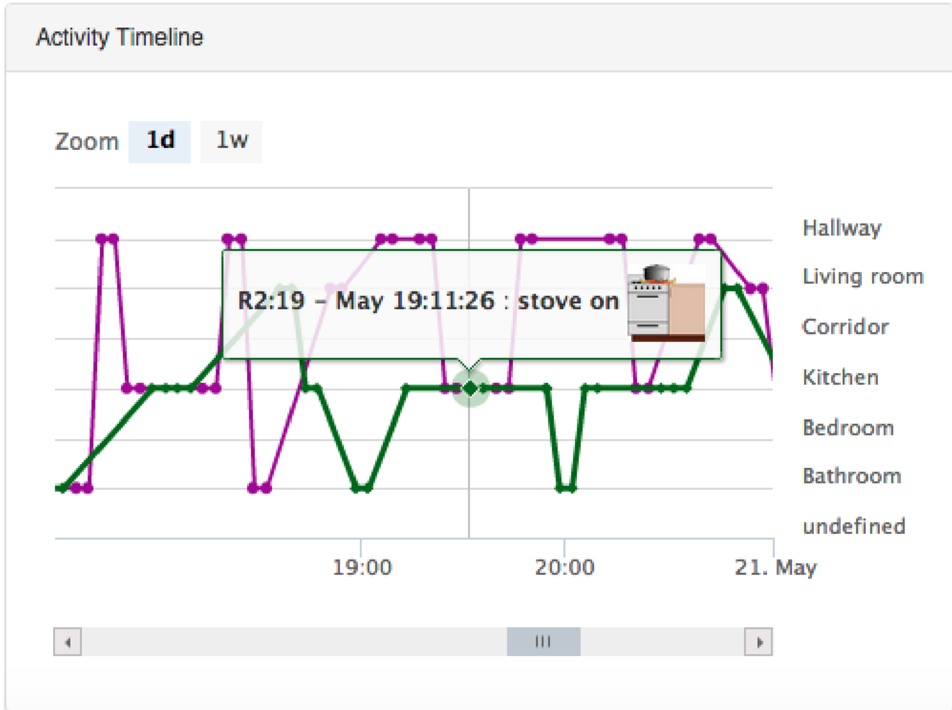

Figure 5.12: Activity timeline graph for 2 residents

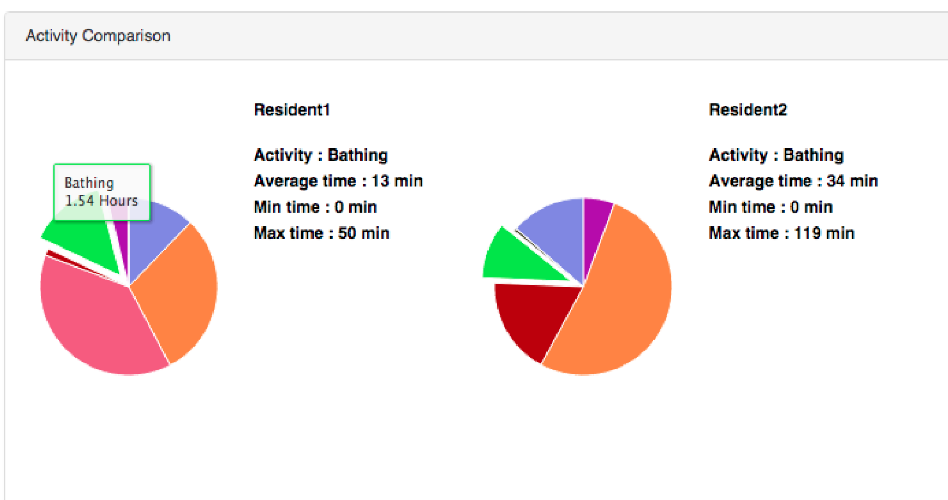

Figure 5.13: Activity pie chart for 2 residents with statistical reports

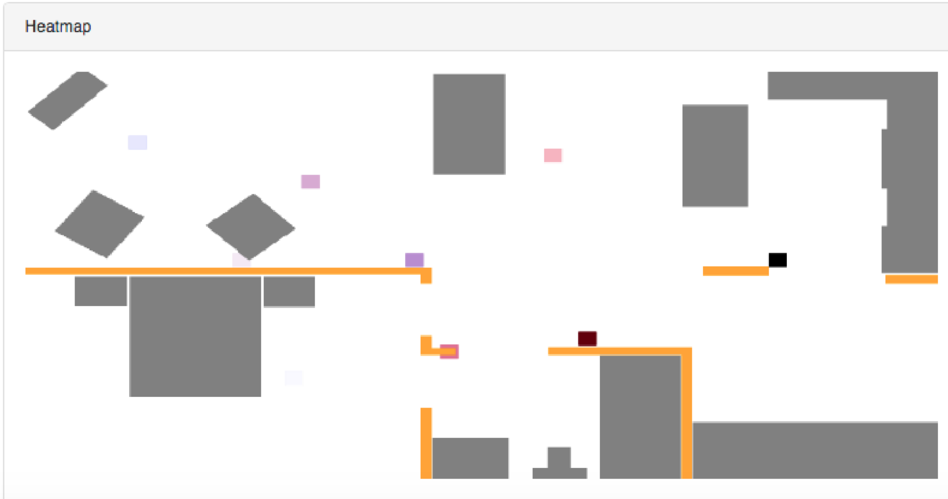

Figure 5.14: Heatmap graph for 2 residents

spent on a specific activity for each resident, which is shown in Figure 5.15.

5.3 Scenario 3: Real-time Visualization (Smart-Condo™ Real-Time Data)

Real-time monitoring is another feature that makes SmartVis as one of the powerful and flexible systems in domain of ambient assisted living monitoring systems. To show the system capabilities, we have devoted the third scenario to visualize the real-time data. For this purpose, we have asked a person to move in the Smart-Condo™ and perform different activities.

5.3.1 View Setup

To view the graphs for this scenario, the user should select the "Smart-Condo Real-time Data" from the dataset tab. As our group is still working on the "ambiguity" issue for multiple residents in real-time mode, this specific type of visualization only works with a single resident. By clicking on the "Draw Graph" button, the dashboard will be filled with 3 types of graphs.

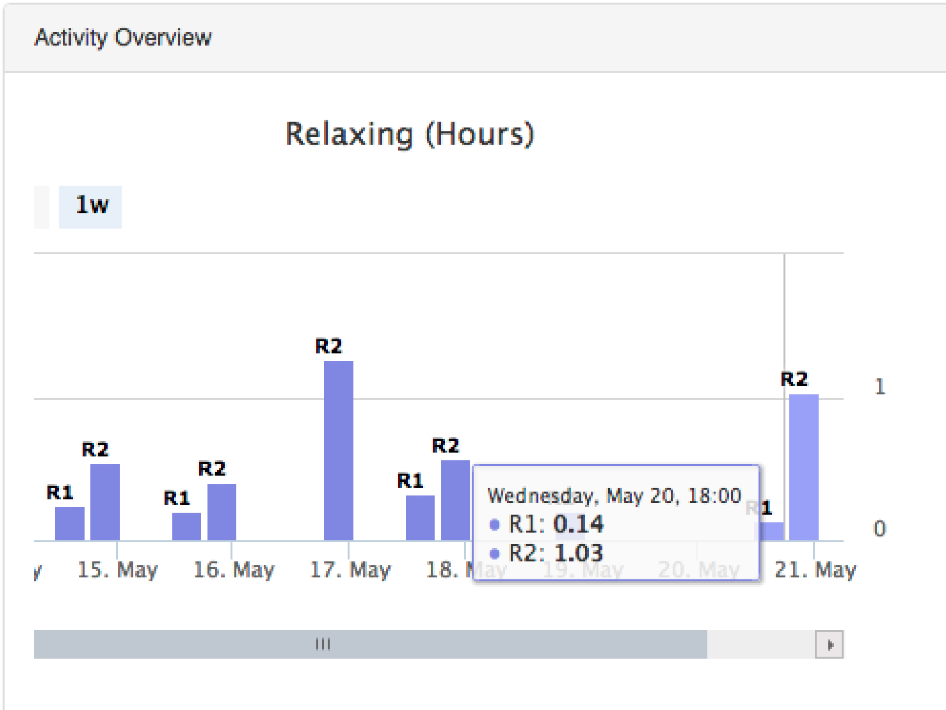

Figure 5.15: Activity overview for 2 residents during a week

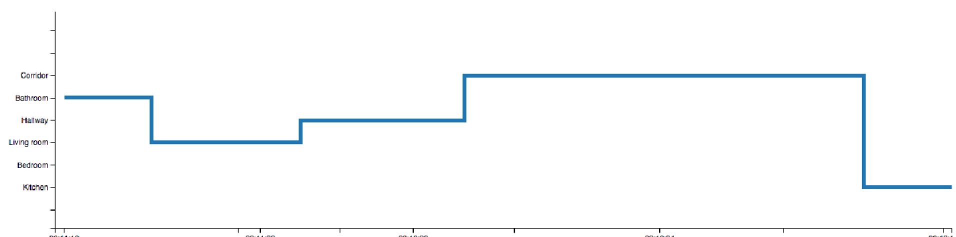

Figure 5.16: Real-time activity timeline for a single resident

5.3.2 Graphs and Views

The first graph is a shifting timeline graph consisting a moving line, which shows the current area of the condo where the person is moving (Figure 5.16). If the user is curious about what the person is doing, he/she can move the cursor over the line and find the corresponding information in the provided tooltip. The next interesting graph is the heatmap, which instantly shows the location of the person on the floorplan of the Condo. Moreover, the user can find out which areas of the Condo are visited the most as the user moving around the house and produce a more colorful heatmap.

Figure 5.17: Two views of the 3D model of the Smart-Condo™ suite and avatar.

Another visualization provided for the real-time data is the virtual world. In this view, caregivers can monitor the avatar's actions, which directly reflect the user activities as they are performed. Figure 5.17 shows two views of the 3D model of the Smart-Condo™ suite and avatar.

Chapter 6

Evaluation

Previous work on smart home visualization has usually focused on particular aspects and limited analysis techniques. Here, we are interested in a more automated, flexible and interactive tool that can be adopted by the user to explore and review multiple aspects of a data slice or compare two slices together. However, this approach, in addition to technical challenges, involves with some practical difficulties. For example, providing appropriate data for the system is still a big challenge. Even though, developing smart homes is an attractive topic these days and many researchers are working in this area, you cannot find many public available datasets. Even, if it is found, it usually comes with missing data. For example, in one dataset you may find “Sensor Events” data for a specific period of time, but the map and footprint of the smart-home, which is needed for the heat-map visualization, is missed. Therefore, our system is evaluated by the Smart-Condo™ simulator, in which we have generated several test cases for individual and multiple residents living in different apartments of a smart building. Moreover, to evaluate functionalities and different components of our system, we have used the CASAS dataset, which is discussed thoroughly in the next section. We also present the results of the studies conducted on these collected data using different capabilities of our system.

6.1 The CASAS Study

An aging population, high cost of health care and the importance of letting individuals aging in their own place for as long as possible, are the three most important reasons for researchers to develop smart environments to emerge and assist with valuable functions such as remote health monitoring. Center for Advanced Studies in Adaptive Systems (CASAS) at Washington State University serves to meet research needs about testing technologies using real data through using smart homes environment located on the WSU Pullman campus.

Dataset Overview: To evaluate our toolkit, we have used a dataset from Washington State University CASAS. Specifically, the dataset has been recorded in the Kyoto testbed, and includes 2,804,813 sensor events collected from WSU apartment for two residents living there during the 2009-2010 academic year. Each sensor event is a row in the dataset including the date and time of when the event occurred, and also the name (e.g. M017) and the status (on/off) of the triggered sensor. Furthermore, the start and end time of various activities performed by a specific resident are annotated at the end of the corresponding sensor event lines. To make the dataset consistent with our intended system input format, we need to perform a preprocessing step, in which we merge multiple rows of the raw dataset into a single one. This row indicates the time stamped activity preformed by a resident. After preprocessing, the new dataset contains 3741 rows of data.

The WSU apartment includes three bedrooms, one bathroom, a kitchen, and a living/dining room. A group of sensors used in the dataset are: motion sensors to detect people's movement in the apartment area; door sensors to measure open-close status of the doors; burner, cold water and hot water sensors to measure the use of stove, cold or hot water respectively; electricity sensor is used to measure the electricity usage; item sensors to check the on/off status of the selected items in the kitchen, and finally temperature sensors to measure the temperature of an specific area. The layout of the apartment is shown in Figure 6.1.

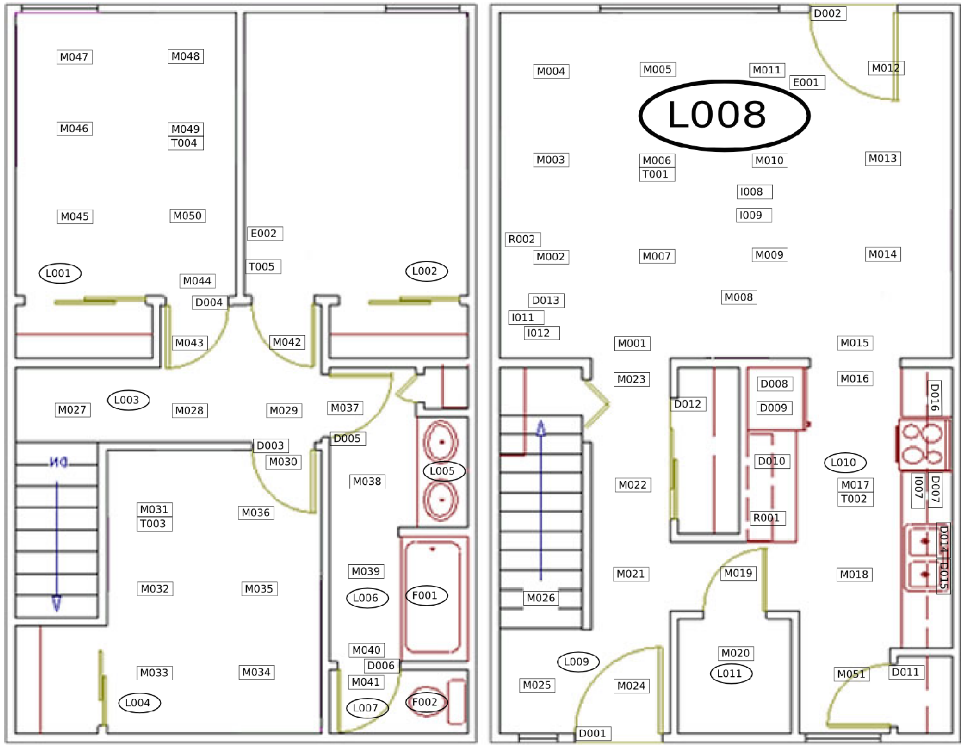

Figure 6.1: The sensor layout of the WSU apartment.

6.2 The Smart-Condo™ Study

The Smart-Condo™ is a comprehensive platform, which provides a variety of services with the goal of improving the healthcare delivery. This one bedroom condo is equipped with movable counters, fully functional kitchen and removable floors for the installation of wireless sensor technology for remote monitoring. The integration of the smart technology, such as wireless sensors for remote monitoring, can improve the life quality for older adults and reduce hospital stays. This technology is used to monitor health-related events and transmit collected data from the embedded sensors in the condo to a central system. Figure 6.2 shows the Smart-Condo™ floorplan [46].

Dataset Overview: As we mentioned before, we typically had several limitations and constraints in accessing smart-home datasets and in general it was very challenging to create an appropriate dataset for the purpose of our analyses. The major problem was creating datasets for multiple residents, as it

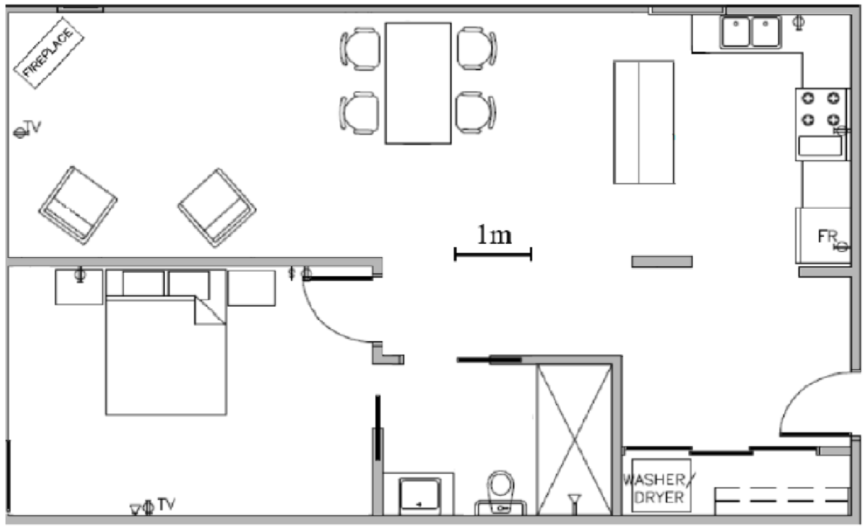

Figure 6.2: The Smart-Condo™ floorplan (grey walls, black doors).

requires a great amount of memory and a significant time to be generated.

Although evaluating with a bigger dataset magnifies the robustness of the system, due to the mentioned problem we had to continue with a smaller dataset to test different aspects of our system. The initial dataset contains sensor events for 80 apartments (40 apartments with 1 occupant, 40 apartments with 2 occupants). In this dataset, the sensor events are generated by the simulator over a 3-day period for each resident.

As we discussed in the previous chapter, with multiple occupancy, the "ambiguity" issue will be inevitable. In other words, in some cases, the simulator cannot detect the occupant who has triggered the sensor event. Therefore, in the dataset, we can see "Null" as residents for a number of rows. In our visualizations, we consider that anyone of the residents could be responsible for the event with the "Null" column.These special sensor events are shown with a different method of display to distinguish between the regular and ambiguous sensor events.

6.3 Evaluation Methods

In our study, we evaluate our toolkit by applying a performance test to check the tolerable load of the system by measuring the response time.

Performance Test: The main aim of performance testing is to measure the actual performance of a web application, evaluate the performance that the system could provide, identify the possible bottlenecks, and finally providing useful tips to fix the problem. In other words, performance testing is used to measure how fast and efficient a software application can complete specified computing tasks. These computing tasks can be categorized into two major groups, where each may be measured using different metrics. The first category is OLTP (Online Transaction Processing), which consists of interactive user activities. To measure how fast a system can respond to requests of the interactive users, response time is used, whereas for non-interactive batch jobs, the metric of throughput is preferred to measure the number of transactions a system can complete over a period of time. [25]

Test Environment: An environment where all tools and hardware are present to perform the evaluation (e.g. performance testing) is called the testing environment. The system that we have done the experiments on is 13-inch MacBook with the following characteristics:

- Operating System: OS X Yosemite (version 10.10.4)
- Processor: 2.9 GHz Intel Core i7
- Memory: 8 GB 1600 MHz DDR3

We should note that each of the above characteristics has a significant influence on the performance testing, so that a little change in any of them may show a great difference on the achieved results.

Performance Metric: As the goal of performance testing is to identify potential risks and improve the system, we should determine the desired features of our application in the early stages of developing the system. In order to achieve the desired features, which often include the user requirements, we need to define criteria to evaluate our work. Response time is the metric we used for evaluation, which means the time taken from one system node to respond to the request of another. In this thesis we claim that our application is an interactive visualization toolkit. As one of the criteria for an application being interactive is that the changes made by the user should incorporated into the visualization in a timely manner, response time is the selected measure for our application evaluation.

Plan and Design Tests: The most common purpose of performance testing is to simulate user scenarios in a realistic mode. In other word, tester should reproduce the user's behaviour while using the system. The test workloads should be very close to real-world scenarios.

To design our application, we considered 6 different scenarios, where in each one, we check the system response time for all the visualization graphs. From scenario 1 to 6, we have increased the input load of the system by adding more (single and multiple) resident's activity information. We have started our experiments by adding 1 single and 1 multiple residents (3 residents in total) to our dataset. In the next scenario, we have tested SmartVis by 5 single and 5 multiple residents (15 residents in total). Scenarios 3 to 6 include 10, 20, 30 and 40 single and multiple residents, respectively; which are randomly selected from the Smart-Condo™ simulator datasets (dataset details can found in section 6.2). To measure the response time, we have used Firebug. Firebug is a development tool integrated with web browsers to edit, debug, and monitor CSS, JavaScript and HTML live in any web page. The results are shown in Table 6.1. The numbers in the table reveal that the response time of the system will obviously increase by adding more load; however this increase is not as significant as the growth of data during the scenarios.

Table 6.1: Performance test measuring response time of 4 visualization graphs. (Numbers are in milli-seconds)

	Scenario	**Overview Activity Graph**	**Activity Pie Chart**	**Timeline Activity Graph**	**Heatmap**
Scenario 1	Individual Mode	9	8	23	9
	Comparison Mode	15	13	39	17
Scenario 2	Individual Mode	9	10	49	18
	Comparison Mode	19	17	100	36
Scenario 3	Individual Mode	13	11	85	30
	Comparison Mode	26	20	173	62
Scenario 4	Individual Mode	21	12	166	59
	Comparison Mode	42	25	228	121
Scenario 5	Individual Mode	31	14	249	77
	Comparison Mode	60	28	489	156
Scenario 6	Individual Mode	37	13	320	111
	Comparison Mode	70	32	642	223

Chapter 7

Conclusion and Future Work

In this thesis, we developed SmartVis as an interactive visualization tool to monitor and analyze older adults activities resided in a smart home. Our work makes the following contributions to this area:

- First, we have developed a prototype application that enables users to explore the sensor data logged from a smart home by employing different graphs and visualizations. This flexible and interactive interface supports users in their task of monitoring elderly daily activities in a non-intrusively manner. The application can be used to explore and analyze smart home datasets based on a set of sensor events stored in a database, where different queries can be made through selecting the patient/s whom the user is interested to see the records over a period of time. (Chapter 4)

- Second, we have designed 4 different types of visualizations, where each one attempts to address one of the user requirements in monitoring an ambient assistive living. These graphs have the capability of showing the selected data for a single patient or compare different patient's activities via the multiple view mode. Moreover, each chart covers a number of features, which makes our system interactive. (Chapter 5)

- Third, we have organized our system by adding smart home configuration files (Appendix A), which makes it very flexible to work with almost any smart home configuration that provides *Sensor Events* collected from the sensors as the input data. To check this functionality, we have used our system in conducting several studies on the generated datasets including samples of CASAS dataset and Smart-Condo™ (simulator and real-time) datasets (Chapter 5).

- Finally, to give empirical support to our system, we compare the performance of the application by defining six different scenarios, where the first scenario has the smallest load of data and the size of the input data is increased by moving forward in the scenarios. Overall, our experimental results reveal that SmartVis performs well for large scales of data. (Chapter 6)

Our initial analyses provide strong evidence that our system is indeed useful in monitoring elderly activities as a web application. The framework potentially can be extended by adding more interactive visualizations. In this regard and as a part of our future work, we are planning to add graphs and reports, which illustrate the health status of the patients. The rendering information will contain the patient's heart beat, blood pressure and so forth. We believe that including these visualizations will be very helpful in exploring and analyzing the current developed graphs.

We also plan to conduct a user study to evaluate the application from the user's point of view. At this level, we couldn't perform the evaluation due to ethical issues. We hope that in near future, we can ask a group of nurses working for a rehabilitation hospital for an interview and apply their helpful suggestions to our system.

Bibliography

[1] Abowd, G.D.: Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM systems journal 38(4), 508–530 (1999)

[2] Adlam, T., Gibbs, C., Orpwood, R.: The gloucester smart house bath monitor for people with dementia. Phys Med 17(3), 189 (2001)

[3] Alexander, G.L., Rantz, M., Skubic, M., Aud, M., Wakefield, B., Florea, E., Paul, A.: Sensor systems for monitoring functional status in assisted living facilities. Research in gerontological nursing 1(4), 238 (2008)

[4] Bashshur, R., Shannon, G., Krupinski, E., Grigsby, J.: The taxonomy of telemedicine. Telemedicine and e-Health 17(6), 484–494 (2011)

[5] Blanchet, K.D.: Remote patient monitoring. Telemedicine and e-Health 14(2), 127–130 (2008)

[6] Bobick, A.F., Intille, S.S., Davis, J.W., Baird, F., Pinhanez, C.S., Campbell, L.W., Ivanov, Y.A., Schütte, A., Wilson, A.: The kidsroom: A perceptually-based interactive and immersive story environment. Presence: Teleoperators and Virtual Environments 8(4), 369–393 (1999)

[7] Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F.: Extensible markup language (xml) 1.0 (fifth edition) (2008), https://www.w3.org/TR/REC-xml, [Online; accessed July-2015]

[8] CASAS: Washington state university (2015), http://casas.wsu.edu/, [Online; accessed June-2015]

[9] Demiris, G., Skubic, M., Keller, J., Rantz, M.J., Parker Oliver, D., Aud, M.A., Lee, J., Burks, K., Green, N.: Nurse participation in the design of user interfaces for a smart home system. In: Proceedings of the International Conference on Smart Homes and Health Telematics, NI Belfast, Editor. pp. 66–73 (2006)

[10] Diehl, M., Marsiske, M., Horgas, A.L., Rosenberg, A., Saczynski, J.S., Willis, S.L.: The revised observed tasks of daily living: A performance-based assessment of everyday problem solving in older adults. Journal of Applied Gerontology 24(3), 211–230 (2005)

[11] Erl, T.: Service-oriented architecture: a field guide to integrating XML and web services. Prentice Hall PTR (2004)

[12] Evans, D.A., Funkenstein, H.H., Albert, M.S., Scherr, P.A., Cook, N.R., Chown, M.J., Hebert, L.E., Hennekens, C.H., Taylor, J.O.: Prevalence of alzheimer's disease in a community population of older persons: higher than previously reported. Jama 262(18), 2551–2556 (1989)

[13] Franklin, M., Zdonik, S.: âĂIJdata in your faceâĂİ: push technology in perspective. In: ACM SIGMOD Record. vol. 27, pp. 516–519. ACM (1998)

[14] Ganti, R., Mohomed, I., Raghavendra, R., Ranganathan, A.: Analysis of data from a taxi cab participatory sensor network. In: Mobile and ubiquitous systems: Computing, networking, and services, pp. 197–208. Springer (2012)

[15] Gatalsky, P., Andrienko, N., Andrienko, G.: Interactive analysis of event data using space-time cube. In: Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on. pp. 145–152. IEEE (2004)

[16] Gavrilov, L.A., Heuveline, P.: Aging of population. The encyclopedia of population 1, 32–37 (2003)

[17] Gil, N.M., Hine, N.A., Arnott, J.L., Hanson, J., Curry, R.G., Amaral, T., Osipovic, D.: Data visualisation and data mining technology for supporting care for older people. In: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility. pp. 139–146. ACM (2007)

[18] Group, N.W.: The application/json media type for javascript object notation (json) (2006), `http://tools.ietf.org/html/rfc4627/`, [Online; accessed July-2015]

[19] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems 29(7), 1645–1660 (2013)

[20] Guinard, D., Trifa, V.M., Wilde, E.: Architecting a mashable open world wide web of things. ETH, Department of Computer Science (2010)

[21] Havranek, E.G., Sharfi, A.R., Nour, S., Motiwala, H., Karim, O.: Low-cost telemedicine. BJU international 107(11), 1701–1702 (2011)

[22] Hussain, S., Erdogen, S.Z., Park, J.H.: Monitoring user activities in smart home environments. Information Systems Frontiers 11(5), 539–549 (2009)

[23] Kientz, J.A., Patel, S.N., Jones, B., Price, E., Mynatt, E.D., Abowd, G.D.: The georgia tech aware home. In: CHI'08 extended abstracts on Human factors in computing systems. pp. 3675–3680. ACM (2008)

[24] LeBellego, G., Noury, N., Viron, G., Mousseau, M., Demong, J.: A model for the measurement of patient activity in a hospital suite. Information Technology in Biomedicine, IEEE Transactions on 10(1), 92–99 (2006)

[25] Liu, H.H.: Software performance and scalability: a quantitative approach, vol. 7. John Wiley & Sons (2011)

[26] Marek, K.D., Rantz, M.J.: Aging in place: A new model for long-term care. Nursing administration quarterly 24(3), 1–11 (2000)

[27] Memon, M., Wagner, S.R., Pedersen, C.F., Beevi, F.H.A., Hansen, F.O.: Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. Sensors 14(3), 4312–4341 (2014)

[28] Mozer, M.: Lessons from an adaptive house. Ph.D. thesis, Architectural Engineering (2004)

[29] Mozer, M.C.: The neural network house: An environment hat adapts to its inhabitants. In: Proc. AAAI Spring Symp. Intelligent Environments. pp. 110–114 (1998)

[30] Mulvenna, M., Carswell, W., McCullagh, P., Augusto, J.C., Zheng, H., Jeffers, P., Wang, H., Martin, S.: Visualization of data for ambient assisted living services. Communications Magazine, IEEE 49(1), 110–117 (2011)

[31] Nick, M., Becker, M.: A hybrid approach to intelligent living assistance. In: Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on. pp. 283–289. IEEE (2007)

[32] North, C., Shneiderman, B.: Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In: Proceedings of the working conference on Advanced visual interfaces. pp. 128–135. ACM (2000)

[33] Ramos, C., Marreiros, G., Santos, R., Freitas, C.F.: Smart offices and intelligent decision rooms. In: Handbook of Ambient Intelligence and Smart Environments, pp. 851–880. Springer (2010)

[34] Rantz, M., Skubic, M., Miller, S., Krampe, J.: Using technology to enhance aging in place. In: Smart Homes and Health Telematics, pp. 169–176. Springer (2008)

[35] Rantz, M.J., Marek, K.D., Aud, M.A., Johnson, R.A., Otto, D., Porter, R.: Tigerplace: A new future for older adults. Journal of Nursing Care Quality 20(1), 1–4 (2005)

[36] Rantz, M.J., Skubic, M., Koopman, R.J., Phillips, L., Alexander, G.L., Miller, S.J., Guevara, R.D.: Using sensor networks to detect urinary tract infections in older adults. In: e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on. pp. 142–149. IEEE (2011)

[37] Rashidi, P., Mihailidis, A.: A survey on ambient-assisted living tools for older adults. Biomedical and Health Informatics, IEEE Journal of 17(3), 579–590 (2013)

[38] Shrestha, A., Miller, B., Zhu, Y., Zhao, Y.: Storygraph: Extracting patterns from spatio-temporal data. In: Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics. pp. 95–103. ACM (2013)

[39] Sjöbergh, J., Tanaka, Y.: Visual data exploration using webbles. WWS pp. 119–128 (2013)

[40] Staggers, N., Kobus, D.: Comparing response time, errors, and satisfaction between text-based and graphical user interfaces during nursing order tasks. Journal of the American Medical Informatics Association 7(2), 164–176 (2000)

[41] Suryadevara, N.K., Mukhopadhyay, S.C.: Wireless sensor network based home monitoring system for wellness determination of elderly. Sensors Journal, IEEE 12(6), 1965–1972 (2012)

[42] Takács, B., Hanák, D.: A mobile system for assisted living with ambient facial interfaces. IADIS Int. J. Comput. Sci. Inf. Syst 2, 33–50 (2007)

[43] Tamura, T., Kawarada, A., Nambu, M., Tsukada, A., Sasaki, K., Yamakoshi, K.I.: E-healthcare at an experimental welfare techno house in japan. The open medical informatics journal 1, 1 (2007)

[44] Thomas, B.L., Crandall, A.S.: A demonstration of pyviz, a flexible smart home visualization tool. In: Pervasive Computing and Communications

Workshops (PERCOM Workshops), 2011 IEEE International Conference on. pp. 304–306. IEEE (2011)

[45] Version, M.: 3.1. 1. edited by andrew banks and rahul gupta. 12 december 2013. oasis committee specification draft 01/public review draft 01

[46] Vlasenko, I.: DEPLOYMENT PLANNING FOR LOCATION RECOGNITION IN THE SMART-CONDO TM: SIMULATION, EMPIRICAL STUDIES AND SENSOR PLACEMENT OPTIMIZATION. Ph.D. thesis, University of Alberta (2013)

[47] Weber, R.H., Weber, R.: Internet of Things. Springer (2010)

[48] Youngblood, G.M., Cook, D.J., Holder, L.B.: Managing adaptive versatile environments. Pervasive and Mobile Computing 1(4), 373–403 (2005)

Appendix A

Configuration Files for Smart-Condo™

Listing A.1 shows different types of sensors embedded in Smart-Condo™ . In this file, each sensor is labeled with a unique ID and name. The tag “type” identifies the type of the sensor, which can be a motion, pressure, switch, humidity, light or current sensor. Another important tag here is the “debounce" time, which shows the delay time involved in switching the signal, which is measured for a group of sensors.

Listing A.1: XML file defining sensor types and their characteristics used in the Smart-Condo™ (Config.xml)

```xml
<?xml version="1.0"?>
<config>
  <mqtt id="0">
    <url>localhost</url>
  </mqtt>
  <VW>
    <XML_remote_channel>e148fae2-275a-4874-8110-9db62a8a989a</
        XML_remote_channel>
  </VW>
  <node>
    <!-- time in seconds for how long to buffer observations (suggested: 0 -
         for testing, 120 - for real deployment) -->
    <min_tx>2</min_tx>
    <!-- how often to send an empty message if there are no observations
    (also affects the time any of these parameters can be changed if there
        are currently no observations)-->
    <max_tx>3600</max_tx>
    <!-- delay between repetitive attempts to connect to the sink (after
        powerup or reset) -->
    <collect_sinks_delay>2</collect_sinks_delay>
    <!-- the number of retries to connect to the sink (after powerup or
```

```xml
    reset) -->
  <collect_sinks_retries>2</collect_sinks_retries>
  <!-- sleep time after a number of unsuccessful retries -->
  <collect_sinks_sleep>30</collect_sinks_sleep>
  <ack_delay>1</ack_delay>
  <ack_retries>30</ack_retries>
</node>
<sensor id="1" desc="passive infrared (standard)">
  <debounce>1025</debounce>
  <type>Motion</type>
</sensor>
<sensor id="2" desc="passive infrared (slight)">
  <debounce>1025</debounce>
  <type>Motion</type>
</sensor>
<sensor id="3" desc="passive infrared (spot)">
  <debounce>512</debounce>
  <type>Motion</type>
</sensor>
<sensor id="4" desc="passive infrared (10 m)">
  <debounce>1025</debounce>
  <type>Motion</type>
</sensor>
<sensor id="5" desc="switch">
  <debounce>1025</debounce>
  <type>Switch</type>
</sensor>
<sensor id="6" desc="Digital FlexiForce (1 pound)">
  <debounce>1025</debounce>
  <type>Pressure</type>
</sensor>
<sensor id="7" desc="Digital FlexiForce (25 pounds)">
  <debounce>1025</debounce>
  <type>Pressure</type>
</sensor>
<!-- 8 = knock, not used for now -->
<sensor id="9" desc="Analog FlexiForce (1 pounds)">
  <sleep_time>4096</sleep_time>
  <!-- diff_threshold depends on particular node and is set in nodes.xml
      -->
  <type>Pressure</type>
  <analog>true</analog>
</sensor>
<sensor id="10" desc="Analog FlexiForce (25 pounds)">
  <sleep_time>4096</sleep_time>
  <!-- diff_threshold depends on particular node and is set in nodes.xml
      -->
  <type>Pressure</type>
  <analog>true</analog>
</sensor>
<sensor id="11" desc="Analog FlexiForce (100 pounds)">
```

```xml
    <sleep_time>4096</sleep_time>
    <!-- diff_threshold depends on particular node and is set in nodes.xml
       -->
    <type>Pressure</type>
    <analog>true</analog>
  </sensor>
  <sensor id="12" desc="Current sensor">
    <sleep_time>4096</sleep_time>
    <type>Current</type>
  </sensor>
  <sensor id="13" desc="Light sensor">
    <sleep_time>4096</sleep_time>
    <type>Light</type>
    <analog>true</analog>
  </sensor>
  <sensor id="14" desc="Humidity sensor">
    <sleep_time>4096</sleep_time>
    <type>Humidity</type>
    <analog>true</analog>
  </sensor>
  <sensor id="15" desc="Pillbox">
    <debounce>1025</debounce>
    <type>Pillbox</type>
  </sensor>
  <sensor id="16" desc="RFID">
    <debounce>1025</debounce>
    <type>RFID</type>
  </sensor>
  <sensor id="17" desc="Virtual">
    <debounce>1025</debounce>
    <type>Virtual</type>
  </sensor>

</config>
```

In Listing A.2 each embedded sensor in the Smart-Condo™ is described by a group of tags. First, we should note that all tags started with "VW" are defined to be used by the virtual world avatar. Second, this file is related to the XML file in Listing A.1 by "sensor" tag. In other words, this tag identifies the type of the sensor. The tag "location" determines coordinates, and "area" represents the region of the associated sensor. "on" and "off" tags are designed to be sent as event status labels to the visualizations via PubNub. The "dominant" tag is a way to recognize the priority of the trigerred event in comparison with other events. For instance, the value of the "dominant" tag for the left recliner is

“ON" in the following XML file, which means sitting on the chair has priority to any other event that may occur concurrently, *i.e.,* movements detected by motion sensors. To state the configuration of RFID readers, tags for the orientational angel (“tetha0"), angle of view (“tetha"), and the sensing range of the reader(“radius”) are considered.

Listing A.2: XML file describing individual sensors embedded in the Smart-Condo™ (Node.xml)

```
<nodes>

   <node id="6" desc="microwave">
      <sensor>17</sensor>
    <VW_location>183.844,52.001,23.400</VW_location>
      <location>9.99,4.46</location>
    <VW_on_command>rotate_;microwave off</VW_on_command>
    <VW_off_command>rotate_east;microwave on</VW_off_command>
      <on>on</on>
    <off>off</off>
      <dominant>OFF</dominant>
      <area>Kitchen</area>
      <size>42,41</size>
      <label>microwave</label>
  </node>

   <node id="7" desc="stove">
      <sensor>17</sensor>
    <VW_location>183.844,52.001,23.400</VW_location>
      <location>9.92,5.34</location>
    <VW_on_command>rotate_;stove off</VW_on_command>
    <VW_off_command>rotate_east;stove on</VW_off_command>
      <on>on</on>
    <off>off</off>
      <dominant>OFF</dominant>
      <area>Kitchen</area>
      <size>47,62</size>
      <label>stove</label>
  </node>

   <node id="8" desc="tv">
      <sensor>17</sensor>
    <VW_location>183.844,52.001,23.400</VW_location>
      <location>1.72,6.2</location>
    <VW_on_command>rotate_;tv off</VW_on_command>
    <VW_off_command>rotate_east;tv on</VW_off_command>
      <on>on</on>
    <off>off</off>
      <dominant>OFF</dominant>
      <area>Living room</area>
```

```
        <size>109,25</size>
        <label>tv</label>
    </node>

    <node id="9" desc="chair">
          <!--pressure sensor in chair-->
      <sensor>9</sensor>
      <VW_location>185.388,57.638,24.218</VW_location>
       <location>6.82,5.25</location>
      <VW_on_command>rotate_east;character_sit</VW_on_command>
      <VW_off_command>character_stand</VW_off_command>
       <on>sit</on>
      <off>stand</off>
      <dominant>ON</dominant>
        <area>Living room</area>
        <size>40,40</size>
        <label>chair</label>
    </node>

    <node id="10" desc="left recliner">
      <!--pressure sensor in a chair (left recliner)-->
      <sensor>9</sensor>
      <VW_location>184.584,56.289,24.500</VW_location>
        <location>2.35,4.12</location>
      <VW_on_command>rotate_north;character_sit</VW_on_command>
      <VW_off_command>character_stand</VW_off_command>
        <on>sit</on>
      <off>stand</off>
      <dominant>ON</dominant>
      <area>Living room</area>
        <size>36,36</size>
        <label>lr</label>
    </node>
    <!--   switches  -->
    <node id="12" desc="cupboard #12">
          <!-- cupboard on top of the stove, 1, 0 - left and right  -->
        <sensor>5</sensor>
        <location>9.75,5.18</location>
      <VW_location></VW_location>
      <VW_on_command></VW_on_command>
      <VW_off_command></VW_off_command>
        <size>10,40</size>
        <on>closed</on>
      <off>opened</off>
      <area>Kitchen</area>
        <label>12</label>
     </node>

    <node id="13" desc="cupboard #13">
        <!-- 1-door (sens_idx=0) and 2-door cupboards (sens_idx=2 left, sens_idx=3 right) -->
```

```
    <sensor>5</sensor>
      <location>8.60,5.75</location>
      <VW_location>186.386337,52.943546,23.498209</VW_location>
    <VW_on_command>rotate_east;cupboard close</VW_on_command>
    <VW_off_command>rotate_east;cupboard open</VW_off_command>
      <size>40,10</size>
      <on>closed</on>
    <off>opened</off>
    <area>Kitchen</area>
    <label>13</label>
  </node>

 <node id="14" desc="cupboard #14">
      <!--2 2-door cupboards around the fridge
      2, 3 - left and right doors of microwave cupboard, 0, 1 - left and right on top of fridge -->
    <sensor>5</sensor>
      <location>9.73,3.8</location>
      <VW_location></VW_location>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
      <size>10,40</size>
      <on>closed</on>
    <off>opened</off>
    <area>Kitchen</area>
      <label>14</label>
 </node>

 <node id="15" desc="cupboard #15">
         <!-- 2 1-door cupboards and 1 2-door
         3 - left 1-door, 2 - right 1-door, 0, 1 - left and right   -->
    <sensor>5</sensor>
      <location>9.83,5.78</location>
      <VW_location>186.386337,52.943546,23.49 8209</VW_location>
    <VW_on_command>rotate_east;cupboard close</VW_on_command>
    <VW_off_command>rotate_east;cupboard open</VW_off_command>
      <size>10,30</size>
      <on>closed</on>
    <off>opened</off>
    <dominant>OFF</dominant>
    <area>Kitchen</area>
      <label>15</label>
 </node>

  <node id="16" desc="fridge">
        <!-- 1 - fridge and 0 - freezer -->
      <sensor>5</sensor>
    <VW_location>183.844,52.001,23.400</VW_location>
      <location>9.92,3.87</location>
    <VW_on_command>rotate_east;fridge close</VW_on_command>
    <VW_off_command>rotate_east;fridge open</VW_off_command>
```

```xml
      <on>close</on>
    <off>open</off>
      <dominant>OFF</dominant>
      <area>Kitchen</area>
      <size>47,50</size>
      <label>fridge</label>
  </node>

  <node id="17" desc="enterance door">
    <!-- entrance door -->
    <sensor>5</sensor>
    <VW_location>181.987,51.571,23.449</VW_location>
    <location>10.46,2.5</location>
    <VW_on_command>door close</VW_on_command>
    <VW_off_command>door open</VW_off_command>
    <on>close</on>
    <off>open</off>
    <dominant>OFF</dominant>
    <size>10,80</size>
    <label>door</label>
      <area>Hallway</area>
  </node>

  <node id="18" desc="sliding door #18">
      <!-- sliding door between the washroom and living room, sensor 2 -->
    <sensor>5</sensor>
      <location>5,2.02</location>
      <VW_location>181.866211,58.277000,23.498211</VW_location>
    <VW_on_command>rotate_north;bathroom sliding door close</VW_on_command>
    <VW_off_command>rotate_north;bathroom sliding door open</VW_off_command>
    <on>closed</on>
    <off>opened</off>
      <obstruction_state>ON</obstruction_state>
    <obstructed_grid>182.081,58.504;182.081,58.004</obstructed_grid>
    <area>Bathroom</area>
      <size>75,6</size>
      <label>sd</label>
  </node>

  <node id="19" desc="sliding door #19">
      <!-- sliding door between the washroom and bedroom, sensor 0 -->
    <sensor>5</sensor>
    <location>4.60,1.92</location>
    <VW_location>181.578018,58.701744,23.498211</VW_location>
    <VW_on_command>rotate_west;bedroom sliding door close</VW_on_command>
    <VW_off_command>rotate_west;bedroom sliding door open</VW_off_command>
    <on>closed</on>
    <off>opened</off>
    <obstruction_state>ON</obstruction_state>
    <obstructed_grid>181.581,59.004</obstructed_grid>
    <area>Bathroom</area>
```

```
    <size>6,57</size>
    <label>sd</label>
  </node>

  <node id="20" desc="flush">
    <!-- flushing the toilet -->
    <sensor>5</sensor>
    <VW_location>181.107849,57.159748,23.498211</VW_location>
    <location>5.87,0.18</location>
    <VW_on_command>rotate_south;flush stopped</VW_on_command>
    <VW_off_command>rotate_south;flush toilet</VW_off_command>
    <on>stop</on>
    <off>start</off>
    <area>Bathroom</area>
    <size>45,13</size>
    <label>toilet</label>
  </node>

   <node id="21" desc="sink">
    <sensor>17</sensor>
    <VW_location>180.492,58.380,23.498211</VW_location>
    <location>9.19,6.27</location>
    <VW_on_command>rotate_south;sink on</VW_on_command>
    <VW_off_command>rotate_south;sink off</VW_off_command>
    <on>on</on>
    <off>off</off>
    <dominant>ON</dominant>
    <area>Kitchen</area>
    <literPerMin>5</literPerMin>
    <size>56,33</size>
    <label>sink</label>
  </node>

   <node id="22" desc="bathroom sink">
    <sensor>17</sensor>
    <VW_location>180.492,58.380,23.498211</VW_location>
    <location>4.72,0.64</location>
    <VW_on_command>rotate_south;washroom_sink on</VW_on_command>
    <VW_off_command>rotate_south;washroom_sink off</VW_off_command>
    <on>on</on>
    <off>off</off>
    <dominant>ON</dominant>
    <area>Bathroom</area>
    <literPerMin>5</literPerMin>
    <size>62,45</size>
    <label>wc sink</label>
  </node>

   <node id="23" desc="bathtub">
    <sensor>17</sensor>
    <VW_location>181.125,55.872,23.498211</VW_location>
```

```xml
    <location>6.67,1.89</location>
    <VW_on_command>rotate_south;bathtub on</VW_on_command>
    <VW_off_command>rotate_south;bathtub off</VW_off_command>
    <on>on</on>
    <off>off</off>
    <dominant>ON</dominant>
    <area>Bathroom</area>
    <literPerMin>10</literPerMin>
    <size>64,132</size>
    <label>bathtub</label>
  </node>

   <node id="36" desc="motion sensor">
    <sensor>3</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>

    <location>5.15,2.34,2.5</location>
    <polygon>4.71,2.99,4.71,3.22,4.53,3.59,5.99,3.59,5.99,1.1,
     4.71,1.1,4.71,1.11,4.68,1.11,4.99,1.92,4.99,2.02,4.71,2.02,
     4.71,2.2,4.7,2.21,4.58,2.21,4.3,2.15,4.3,3.13,4.44,3.13,4.58,2.99</polygon>
    <area>Corridor</area>
  </node>

   <node id="41" desc="motion sensor">
    <sensor>3</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
    <location>1.25,5.04,2.5</location>
    <polygon>0.4,6.29,0.4,3.8,2.09,3.8,2.09,6.29</polygon>
    <area>Living room</area>
  </node>

  <node id="42" desc="motion sensor">
    <sensor>3</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
      <location>3.35,4.44,2.5</location>
      <polygon>2.1,5.29,4.59,5.29,4.59,3.6,2.1,3.6</polygon>
      <area>Living room</area>
  </node>

  <node id="43" desc="motion sensor">
    <sensor>3</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
      <location>6.15,4.84,2.5</location>
      <polygon>4.9,5.69,7.39,5.69,7.39,4,4.9,4</polygon>
      <area>Living room</area>
  </node>
```

```xml

<node id="44" desc="motion sensor">
  <sensor>3</sensor>
  <VW_on_command></VW_on_command>
  <VW_off_command></VW_off_command>
    <location>9.35,4.24,2.5</location>
    <polygon>8.5,5.49,10.19,5.49,10.19,3.14,9.95,3.14,
    9.94,3.13,9.94,3,8.53,3,8.61,3.12,8.61,3.26,8.6,3.27,8.5,3.27</polygon>
    <area>Kitchen</area>
</node>

<node id="45" desc="motion sensor">
  <sensor>3</sensor>
  <VW_on_command></VW_on_command>
  <VW_off_command></VW_off_command>
  <location>9.15,1.44,2.5</location>
  <polygon>7.9,2.29,10.39,2.29,10.39,0.6,7.9,0.6</polygon>
  <area>Corridor</area>
</node>

<node id="46" desc="motion sensor">
  <sensor>4</sensor>
  <VW_on_command></VW_on_command>
  <VW_off_command></VW_off_command>
  <location>3.05,1.54,2.5</location>
  <polygon>1.8,2.39,4.29,2.39,4.29,0.7,1.8,0.7</polygon>
  <area>Bedroom</area>
</node>

<node id="47" desc="motion sensor">
  <sensor>3</sensor>
  <VW_on_command></VW_on_command>
  <VW_off_command></VW_off_command>
  <location>0.95,1.24,2.5</location>
  <polygon>0.1,2.49,1.79,2.49,1.79,0.01,0.1,0.01</polygon>
  <area>Bedroom</area>
</node>

 <node id="48" desc="motion sensor">
  <sensor>4</sensor>
  <VW_on_command></VW_on_command>
  <VW_off_command></VW_off_command>
  <location>7.25,3.14,2.5</location>
  <polygon>6,3.99,8.49,3.99,8.49,3.41,7.83,3.26,
  7.83,3.12,8.49,3.08,8.49,2.3,6,2.3</polygon>
  <area>Hallway</area>
</node>

 <node id="60" desc="rfid reader">
  <sensor>16</sensor>
```

```
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
    <radius>1</radius>
    <location>2.45,3.22</location>
    <tetha>90</tetha>
    <tetha0>90</tetha0>
    <area>Living room</area>
  </node>

   <node id="61" desc="rfid reader">
    <sensor>16</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
    <radius>1</radius>
    <location>4.55,3.22</location>
    <tetha>90</tetha>
    <tetha0>90</tetha0>
    <area>Living room</area>
  </node>

   <node id="62" desc="rfid reader">
    <sensor>16</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
    <radius>1</radius>
    <location>4.98,1.89</location>
    <tetha>90</tetha>
    <tetha0>354.61</tetha0>
    <area>Bathroom</area>
  </node>

   <node id="63" desc="rfid reader">
    <sensor>16</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
    <radius>1</radius>
    <location>6.47,2.01</location>
    <tetha>90</tetha>
    <tetha0>90</tetha0>
    <area>Hallway</area>
  </node>

   <node id="64" desc="rfid reader">
    <sensor>16</sensor>
    <VW_on_command></VW_on_command>
    <VW_off_command></VW_off_command>
    <radius>1</radius>
    <location>8.6,3.24</location>
    <tetha>90</tetha>
    <tetha0>73.999</tetha0>
    <area>Kitchen</area>
```

```
  </node>

  <node id="70" desc="pillbox">
    <!-- pill box -->
    <VW_on_command>pillbox close;</VW_on_command>
    <VW_off_command>pillbox open;</VW_off_command>
    <on>closed</on>
    <off>opened</off>
    <sensor>15</sensor>
    <dominant>OFF</dominant>
  </node>

</nodes>
```

Appendix B

PubNub Publisher/Subscriber

Listing B.1 depicts the code used for subscribing to the messages published by the localizer. The first step is to initialize a PubNub connection using the “publish_key” and the “subscribe_key”. PubNub applications works with 2 categories of keys: First, the keys that are privately generated for each application. Second, a key called “demo” that works with any application. Here, to keep the security, we have placed the “demo” keys as our publishing and subscribing keys. The next important item that should be the same on both sides is the channel name. For our application, we chose the “smartcondo” as the channel name. Each message published by the localizer is a JSON object including 6 fields of data: resident id, index number indicating a unique id for each message, description, area and the coordinates of the occurred event. Listing B.2 shows how these messages are created and published.

Listing B.1: JavaScript code describing PubNub subscriber

```
var pubnub = PUBNUB.init({
  publish_key: "demo",
  subscribe_key: "demo"

});


pubnub.subscribe({
  channel: "SmartCondo",
  message: function(m){
    pubnubCallback(m);
    console.log(m);
  },
  error: function (error) {
```

```
    console.log(JSON.stringify(error));
  }
});
```

Listing B.2: Java code describing PubNub publisher

```
public void publish(String resident, String area, String event, Point3D pos, long ts, int counter){
        Pubnub pubnub = new Pubnub("demo", "demo");
        JSONObject columns = new JSONObject();
        Timestamp current = new Timestamp(ts);
        Date date = new Date(current.getTime());
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd'\t\t'HH:mm:ss");
        try {
            columns.put("Index", counter);
            columns.put("resident", resident);
            if(area == null){
                columns.put("area", saveArea);
            }else{
                columns.put("area", area);
                saveArea = area;
            }
            columns.put("event",event);
             if(pos == null){
                columns.put("xc",savePos.x);
                columns.put("yc",savePos.y);
            }else{
                columns.put("xc",pos.x);
                columns.put("yc",pos.y);
                savePos = pos;
            }
            columns.put("x",simpleDateFormat.format(current));

        } catch (JSONException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        pubnub.publish("SmartCondo", columns, new Callback(){
        public void successCallback(String arg0, Object arg1){
        }
        public void errorCallback(String channel, PubnubError error) {
        }

        });
  }
```