8-26-2015 12:00 AM

Gamification Framework for Sensor Data Analytics

Alexandra L'Heureux
The University of Western Ontario

Supervisor
Dr. Miriam Capretz
The University of Western Ontario

Graduate Program in Electrical and Computer Engineering
A thesis submitted in partial fulfillment of the requirements for the degree in Master of Engineering Science
© Alexandra L'Heureux 2015

GAMIFICATION FRAMEWORK FOR SENSOR DATA ANALYTICS

(Thesis format: Monograph)

by

Alexandra L'Heureux

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Acknowlegements

I would like to thank my supervisor Dr. Miriam Capretz for all of her support over the last few years. She motivated me to get involved with research during the third year of my undergraduate degree and has since been a motivation to pursue a career in academia. The accomplishment of this thesis would not have been made possible without her support and guidance. Thank you for always supporting me and believing in my capabilities even through difficult times.

I would also like to thank my research team for their support and inspiring ideas: Wilson Higashino, Sara Abdelkader, Mauro Ribeiro, Dennis Bachmann, Daniel Berhane Araya and especially Dr. Katarina Grolinger who has taken countless hours out of her schedule to review my work and help me shape my ideas. I am immensely grateful for everything you have done for me, thank you.

I would also like to thank the team at Powersmiths for helping me develop my research ideas and providing me with all the tools necessary for the success of my thesis. I am very appreciative of your involvement and support in this project.

This thesis would also not have been made possible without the support of my friends and family. I would especially not be where I am today without the support of my partner and best friend Melissa. Thank you for always being there for me through these crazy few years. You have been an indescribable source of support and strength and I could never thank you enough for that. I love you! To my mother France and brother Cyril, thank you for your unconditional love and support through this crazy and often stressful adventure, I love you.

Lastly, I would like thank my dad Jacques. Thank you for teaching me the value of education and hard work and for always believing in me even when I did not. You showed me how to keep my head up and that I could achieve anything I set my heart out to do. I am so sorry that you were unable to witness what I have accomplished over the last 6 years, I love you and miss you everyday. I dedicate this thesis to you papa.

Abstract

Data in all of its form is becoming a central part of our existence. It is being captured in every facets of our everyday life: social media, pictures, smartphones, wearable devices, smart building etc. One of the main drivers of this Big Data Revolution is the Internet of Things, which enables inert objects to communicate through a multitude of sensors. The data amassed fuels a thirst for information, the extraction of such knowledge is rendered possible through data analytics techniques.

However, when it comes to sensor data our large-scale ability to perform analytics is highly limited by the difficulties associated with collecting sensor data labels. Current crowdsourcing platforms historically used to gather labels are unable to process sensor data due to its low level nature and its reliance upon contextual information. The solution proposed in this thesis enables the deployment of a crowdsourcing platform for sensor data. This research presents a novel solution to acquire sensor labels by leveraging the power of crowdsourcing using gamification. The work in this thesis describes not only a framework that facilitates the capture of sensor data label through a flexible gamification architecture but also a solution that outlines the mechanics required to integrate gamification in a variety of contexts. Additionally, the framework is designed in a flexible manner to support any type of sensor data given that humans can readily interact with them. Additionally, the work presented describes and supports both real time and historical data analytics through the captured data and associated labels.

This work was successfully evaluated in the context of a case study performed in conjunction with an industry partner. The gamification implementation was tested for a number of electrical sensors. Real time and historical data analytics were successfully performed with the use of the framework. The robustness of the solution was evaluated though the injection of invalid data and the result showed that the framework is effectively capable of reducing the level of noise in the data labels.

Keywords: Sensor Data, Data Analytics, Data Labelling, Crowdsourcing, Gamification

Contents

List of Figures

List of Tables

List of Abbreviations

IoT	*Internet of Things*
KNN	*K Nearest Neighbours*
BSS	*Between Sum of Squares*
TSS	*Within Sum of Squares*

Chapter 1

Introduction

The big data revolution vows to turn data into actionable knowledge through big data analytics techniques. Businesses are capturing and storing more and more data with hopes to extract great amounts of valuable information. Indeed, when it comes to products and customer data, that promise of valuable insights is often fulfilled. However in regards to less defined datasets such as those in provenance of sensors, the analysis can be much more complex and the extraction of valuable intel much more difficult.

The Internet of Things (IoT) [2] is an ecosystem powered by sensors and microchips to enable the connection and communication amongst real-life objects, environments and people. Through this network of things, sensors and devices such as household appliances, cars, health devices and even buildings are capturing and exchanging enormous amounts of data and therefore fuelling the big data movement.

However, the data captured from human interactions by sensors is very different from other data collected by humans. Indeed images, social media data and usage information are human interpretable in the capacity that an untrained user can correctly identify what the data represents. Conversely, sensor data cannot typically be easily reconciled and interpreted without prior domain and contextual knowledge.

The functionality of the IoT, depends upon four fundamental steps as suggested by Swan [3] : data creation, information extraction, meaning-making, and action-taking. Therefore, extracting information or knowledge from sensor data is also a crucial piece of the IoT scenario. Data analytics techniques and algorithms can be used to perform such tasks.

Despite that, the human interaction data currently being captured by sensors is often only processed and analyzed by field experts and consequently does not often allow for data analytics to be performed at large scales. The capture of the activities surrounding sensor data readings is essential to the acquisition of sensor data labels as it is the preferred way to establish the truth value or label of a reading. The lack of labels or of contextual information surrounding sensor readings is one of the root causes of the impracticability of data analytics on sensor data.

Therefore there exists an impending need for a sensor data analytics framework that would enable the analysis of sensor data. The solution needs to provide a means of acquiring and labelling sensor data in a flexible manner such that it can be utilized for any type of sensors with which users interact. This thesis will present a solution to such a problem.

1.1 Motivation

Historically, when researchers or users have been working on performing data analytics on datasets they would make use of various techniques. A large number of those methods fall under the supervised machine learning algorithms umbrella [4]. Those algorithms rely upon data labels to learn patterns and characteristics from a dataset and subsequently infer knowledge such as classifications or predictions. In order to obtain such labels and build their datasets, researchers would often use crowdsourcing services [5, 6, 7].

Crowdsourcing can be described as the process of using large groups of various individuals to perform specific tasks. In the context of machine learning, those tasks would be to ask users to identify specific data such as images and the result of the identification would serve as the labels. Mechanical Turk [8] is an example of such a framework, it allows for researchers to post tasks to be performed and in exchange for their participation users receive a financial compensation. Using those frameworks is particularly effective for datasets where humans are much more effective than computers at performing a task. Unfortunately, when it comes to sensor data, typical crowdsourcing frameworks are ineffective due to the poor human readability of the data.

It is quite challenging to apply a label on sensor readings and identify exactly what interactions may have been measured. In certain cases, contextual information may allow for the detection of abnormal behaviours or anomalies [9], however when it comes to identifying or classifying what is being measured, additional information surrounding the readings is required.

Sensor data is most helpful when placed in context with information regarding what is being measured. Human activity is a context of primary importance, as it jointly describes not only a person's intention but also the current status of objects [10] which can be used to label sensor data. Although, in order to gather such information, it would be required to know exactly how these sensors are being used and what is happening at the time of the reading. Existing labelling techniques suggest both real time and post processing labelling approaches. In spite of that, it is often not practical or accurate to obtain this information in after the fact, this type of labelling should occur in parallel with the capture of the data.

One of the main challenges when it comes to sensors is that humans are often unable to manually label the data, leaving current techniques unavailing. An incredible amount of sensor data is gathered and saved everyday but there exists a limited amount of options to leverage intelligence from it. For example, electrical sensors in buildings are capturing copious amounts of readings but limited knowledge can be extracted through data analytics without first capturing some labels for the data. Existing sensor data labelling techniques are costly both in terms of financials and time. New means of gathering labels for sensor data are required and the need for such techniques provides motivation for this work.

Additionally, in the cases of sensor data, although the potential range of readings may be somehow finite, an almost infinite combination of attributes is possible. In other words, as new events and activities are being measured by the sensors, the model initially trained may no longer be accurate. In order to adequately perform real time analysis of data, a mechanism is required to be put in place to quickly capture and label those new events. The new labelling solution must allow for easy and inexpensive addition of labels. This indicates that a static model may not provide the best accuracy and that introducing some interactivity and responsiveness may be key to a successful methodology. This challenge motivated the integration of gamification within this work. Gamification, which is defined as the integration or addition of game elements to non-related contexts [11], would be used not only to heighten the participation and interest of the users but also to allow a fast and effective way to acquire new labels.

Furthermore, current crowdsourcing solutions are not applicable to sensor data. Researchers in the areas where the use of labelled sensor data is primordial are either forced to build their own datasets through various expensive means, or they must make use of the limited amount of public datasets available. The difficulty of generating labelled data for research purposes also served as motivation for this thesis. A fast, cheap and effective method to robustly build a set of sensor data through crowdsourcing is needed for the research community.

The main goal of this research is to address these challenges and issues related to the collection of sensor data by leveraging the power of gamification.

1.2 Contribution

This thesis contains various contributions, however, only the main contributions of our work will be discussed in this section.

First of all, the prime contribution of this thesis lies in the architecture of the gamification framework itself, through its modularity, adaptability and flexibility.

This framework provides a complete end to end solution for sensor data analytics, from the gathering of data, to its labelling, and lastly through its analytical capabilities. It proposes a robust new sensor data labelling technique which leverages the power of gamification to capture human activities and assign them as sensor labels. It demonstrates how human activities can be translated into sensor events of interest.

As opposed to the work found in literature, which tends to only focus on mobile sensors [12, 13], this framework has the ability to quickly and effectively adapt to support a variety of sensors. It was designed to work with any type of sensors, whether mobile or permanently installed in fixed locations. It supports both the analysis of real time and historical sensor events.

The work presented here does not only address the need for easily accessible labelled sensor data, it also enables researchers to request labels for specific types of sensor data. The framework therefore enables not only data analytics for end users but also provides a platform for researchers to crowdsource any sensor related task. This was not previously possible with current existing crowdsourcing solutions. Additionally, the framework enables end users to leverage the newly acquired data to perform analysis of both real-time and historical data.

Lastly, the design of the gamification component of the framework is novel in the fact that it provides the required architecture to integrate gamification within our framework without being directly tied to any particular gamification implementation. Gamification is not only used as a means to achieve a goal, it is a central component to the architecture. The work presented here presents a strategy to integrate gamification directly within the design of a solution. It provides a blueprint to a successful implementation of gamification while remaining completely flexible in the way it is implemented.

This framework could be implemented with any game or context as long as the outlined requirements are met and design followed. Other solutions found in the literature were tied to a specific game implementation and showed gamification as a secondary piece of the design [14, 15]. They did not highlight nor show how the gamification was tied and integrated within the design itself.

In summary, this thesis presents a solution to robustly and effectively label sensor data while enabling sensor data crowdsourcing. It also proposes an architecture to integrate gamification in a flexible and adaptable manner. The results of the evaluation show that the framework can successfully label sensor data with a low amount of noise. The real time and historical data analytics functionalities are also positively demonstrated.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows:

- Chapter 2 will provide a combination of some background information and a literature review of the current data labelling techniques and crowdsourcing gamification frameworks. In the first part, an introduction to the various technical terms and concepts used throughout this thesis will be provided as background information for our work. Secondly, a review of the data labelling techniques commonly used to label sensor data will be provided along with a review of academic work in the area of sensor crowdsourcing. Thereafter, an introduction to gamification as a paradigm for users motivation and engagement will be provided in conjunction with a state of the art review of the combination of gamification and crowdsourcing. Finally, the contribution of this thesis will be re-iterated in the context of existing work.

- Chapter 3 is the main contribution of this thesis. It will present the architecture of the gamification framework for sensor data analytics. The chapter is decomposed in sections each corresponding with one of the six main components of the frameowork: crowdsourcing dashboard, gamification, sensor interface, event detection, event labelling and analytics. First, an overview of the overall framework including its purpose and functionalities will be discussed along with a review of the use cases it addresses. Thereafter, each of the modules will be introduced, their purpose discussed and their design presented. Lastly a summary of the contribution will be provided.

- Chapter 4 presents a general implementation of the gamification framework. It introduces an implementation for each of the six main components presented in Chapter 3. The details of each of the components will also be discussed and the relations between the architecture and the implementation highlighted. Lastly, a summary of the contribution of the implementation will be presented.

- Chapter 5 depicts the evaluation of the gamification framework. It first introduces a case study performed using the implemented framework and secondly presents the methodology used to evaluate the framework. The results of the evaluation of the various components will also be presented along with a short discussion demonstrating the success of the solution.

- Chapter 6 provides the conclusion of our work along with a discussion regarding future work possibilities.

Chapter 2

Background and Literature Review

This chapter serves a dual purpose: first it introduces the various terms related to the topics discussed in this thesis and secondly it provides an overview of the existing work in regards to gamification and sensor data labelling. More specifically, the literature review will focus on the various existing frameworks built for gamification and crowdsourcing. The research gap addressed by this thesis will also be discussed.

2.1 Concept Introduction

This section will define and discuss the following concepts: data labels and analytics, clustering, sensor data and sensor events. These are essential to the understanding of the framework described herein. They serve as a foundation for the understanding of this work.

2.1.1 Data Labels and Analytics

Data labels are defined as a representation of the ground truth or gold standard [16] of a data sample, that is the accurate value of what a sample represents. Having access to the ground truth or data label of a data reading is critical for data analytics due to its reliance upon machine learning techniques. When performing classification, data labels are referred to as classes.

Data labelling is a critical component of supervised machine learning because this entire class of algorithms is entirely dependent upon labels in order to learn and extract knowledge from data. The performance of the machine learning algorithms is directly related to the quality of the labelled dataset [17]. On the other hand, unsupervised machine learning techniques aim at extracting data patterns or discovering similarities from data without having prior access to the labels [18]. However, the data labels are still important in order to validate the efficiency and accuracy of unsupervised algorithms. Therefore, proper data labelling is critical to successful data analytics. Table 2.1 presents the various machine learning algorithm categories and their associated data requirements.

Algorithm Category	*Data Label Requirement*
Unsupervised	Requires data labels to validate the accuracy of the algorithms. Labels are also used to gain additional knowledge from the results such as names for data clusters.
Supervised	Requires data labels to learn from the dataset and tune internal parameters. A substantial data set is required both for learning and validating the algorithm's performance.
Semi Supervised	Requires some specifically chosen data labels from which it learns, these algorithms require far less labels than supervised algorithms. Data labels are also needed to validate the performance.

Table 2.1: Machine Learning Algorithms and their Labelling Requirements.

Typically, datasets are labelled using human annotation performed by professionals such as annotators, field expert, raters, observers, labourers, onlooker, or judges [19] depending on the type of data to be labelled. However, due to human imperfections and data variability labels are not always absolute. There exists four main categories of data labels presented in Table 2.2 [20].

Label Type	*Definition*
Hard Label	Absolute confidence in the data label. The data is associated to one class, all labellers agreed on the assignment.
Soft Label	Varying degree of confidence in various data classes. This may occur when specialists or labellers are not in agreement with the label.
Noisy Label	A label that may contain some erroneous values. For example a sample labelled with the wrong class [21].
Multi-Label	A sample that is assigned multiple labels.

Table 2.2: Types of Data Labels.

Labelled datasets are very important to perform data analytics because labels are necessary to accurately train algorithms, validate assumptions and verify analytical results. The data that they contain must be as accurate as possible to ensure that the results obtained are reliable. Ideally, a dataset would be composed exclusively of hard labels with as little noise as possible. However, there exist many issues related to the acquisition of sensor data labels. Those challenges mainly stem from the low level nature of the data itself. For example the labelling process is much more involved due to the abstractness of the data and therefore requires some contextual information surrounding the readings. These challenges will be discussed more in depth in the upcoming sections.

2.1.2 Clustering

Clustering is a technique that aims at grouping data into clusters based on some beliefs of similarity [22]. As an unsupervised technique, clustering can therefore be used to get some insights on the similarity of the data being observed. In the context of data labelling, clustering is often used to assess the quality of the data labels and remove noisy labels [23]. There exists multiple categories of clustering methods used for data analytics, they will be presented in Table 2.3 [24].

Category	*Idea*
Hierarchical	Data objects are combined into groups and those groups into other groups creating a hierarchy, which can be visualized as a dendogram.
Partitioning relocation	Divides the data objects into subsets and based on a relocation scheme, data objects iteratively get reassigned to different clusters.
Density-based paritioning	Assigns data cluster based on the density and connectivity between data points and regions.
Grid-based	Data space is translated into a grid then space partitioning techniques are applied to cluster the data.
Co-occurence	Used for categorical data, it relies on the idea of transactions and uses co-occurence matrices to perform clustering.

Table 2.3: Data Clustering Methods.

However, the *k*-means method, which falls under the partitioning relocation category, is by far the most used in the scientific community [25]. The idea behind *k*-means clustering is to assign a data point to the cluster with the closest mean. The objective function of *k*-means is to minimize the squared error between the empirical mean of a cluster and the points in the cluster. The algorithm can be described by the following steps [26] :

1. Select a number of clusters K in which you wish to partition the data.
2. Make an initial partition with K clusters, with either random or selected cluster centroids.
3. Calculate new cluster centers by computing the mean of each cluster.
4. Re-reassign each data point to its closest cluster centre.
5. Repeat steps 3 and 4 until the clusters are stable; meaning that the points remain in the same cluster.

The accuracy of the *k*-means clustering can be measured using the BSS/TSS ratio [27]. The BSS/TSS ratio is simply the sum of squares between the groups over the total sum of squares for all the observations, the *k*-means algorithm aims at maximizing this ratio to be as close to 1 as possible.

Additionally, if the ground truth of each cluster is known, clustering algorithms, such as *k*-means, can also be used to perform classification tasks [22].

Another well known algorithm that can be used to perform clustering and classification is the *k*-nearest neighbour or knn algorithm [28]. The idea behind this algorithm is that if a data point belongs to a certain cluster, its *k*-nearest neighbours should also belong to that same cluster. This algorithm is considered to be a supervised algorithm as the class or cluster of some data points are needed to verify if the neighbours are within the same cluster.

2.1.3 Sensor Data

Sensor data varies from other data types by its very nature. Sensor data is often defined as spatio-temporal data which means it contains locational, temporal and numerical data [29]. Sensor data is derived from signals and is therefore abstract in comparison to other types of data that may simply contain an object such as image data.

In terms of sampling frequency, sensor data is generally separated in three categories, they are presented in Table 2.4 [30].

Data Type	*Definition*
Simple Sensor Data	Numerical sensor value sent periodically or on request.
Continuous Sensor Data	Data is sent continuously either in a summarized form or through data capture at specified intervals.
Sensor Event Data	Data is sent when the values are over a predefined threshold.

Table 2.4: Categories of Sensor Data.

However, nowadays with the advent of the IoT [2] most sensors are designed to work with applications requiring continuous sensor data with real time processing capabilities [31]. Therefore, labelling techniques or data analytics frameworks should be designed to support such data.

2.1.4 Sensor Events

In the field of data mining and analytics, an event is described as a “happening of interest”[32]. This could be the accomplishement of a task, if tasks are what we are pursuing.

Table 2.5 presents the definition of various events nomenclature [33]. The definition of the various terms used in relation with sensor events will be defined to facilitate the understanding of this thesis.

Event Semantic Term	*Definition*
Occurrence	Relates to the event taking place.
Detection	Relates to when the event is perceived by the system.
Atomic events or primitive events	An event considered indivisible which either does or does not occur. It is instantaneous and significant [34].
Composite event or complex event	An event composed of either a combination of atomic events or of the sequence of execution of atomic events. They are often described as the result of a reaction rule [34].
Point event	An atomic event that occurs at a single point in time.
Interval event	An atomic event that occurs over an interval of time.

Table 2.5: Sensor Event Semantic Definitions.

Events can be used to provide a description of what is currently being measured by sensors and can therefore serve as annotation or labels for sensor data. Events may also be referred to as activities.

Due to their nature, in literature events are often treated as anomalies [35]. Much work has been performed in the field of sensor network to use anomaly detection techniques to detect events [36, 37]. However, events in the context of this thesis do not meet the definition of anomalous events which are defined has occurrences that are different from typical patterns [38]. In this framework events are not considered to have abnormal patterns, they are instead events of interest, corresponding to the accomplishment of a task. In our work, the readings directly corresponding to an event shall not look anomalous because they are captured as a result of normal device usage.

Now that a basic definition of key terms has been provided, the various techniques and frameworks upon which this thesis was constructed will be presented.

2.2 Sensor Data Labelling Techniques

This section serves as an introduction to the various data labelling techniques used in various sensor data annotation activities. These techniques will be discussed, as well as, the advantages and disadvantages of each considered.

As opposed to other types of data, sensor data is complex to annotate due to the ambiguous and suggestive nature of each reading. Data labels take different forms depending on the type of data we are performing analysis upon; some labels may be absolute and others may be more suggestive. For example, labelling an image of a banana will be absolute because no matter which observer looks at the image, the consensus should be that the image represents a banana. However, for a sensor reading the data is presented like a signal, it is more abstract and open to interpretation.

Due to the low level nature of sensors, it is very complex even for domain experts to describe what is being measured by the sensors. For example, accelerometers are used within a variety of devices. However, it would be extremely difficult if not impossible for domain experts to distinguish between a reading representing motion captured by a wearable device of someone running or that of someone wearing the same device dancing. The expert will be able to provide some insights such as the fact that the device was moving and potentially rotating, but would not be able to identify what motion was performed or exactly what was being measured, simply based on low level numerical data. As a result, sensor data is often annotated and labelled with the activity surrounding the data reading. Activity recognition is a way of using sensors to identify activities and provide sensor context awareness to diverse applications [39].

The work presented in this thesis will use the activity performed surrounding the sensors as the ground truth or label for each sensor reading. This is due to the influence of the IoT which is driving sensor data analytics.

Sensor data labelling is an expensive endeavour and each of the techniques chosen to apply labels must make a compromise between accuracy of labelling and cost. Figure 2.1 shows the various methodologies used to label sensor data. Each of the techniques shows its cost/accuracy tradeoff as well as if it is an online or offline method, that is whether it is performed in real time or through post-processing. A description of the techniques and a review of their advantages and disadvantages will also be presented next [1].

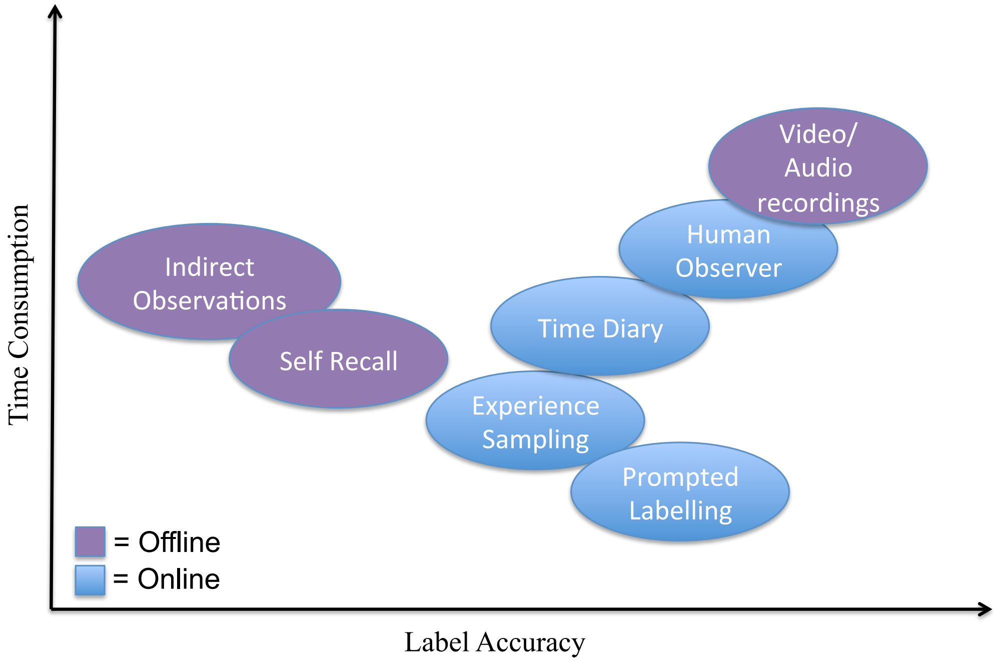


Figure 2.1: Diagram of Annotation and Labelling Techniques. [1]

- Indirect Observation
  - Indirect Observation is labelling based on the data itself without any contextual information. Expert would look at the sensor reading without observing the users

  and appose label based on their expertise.

- Self Recall
  - The self recall technique is performed after the completion of measurements and is therefore considered an offline technique. Users are asked to go back on their day to identify which activities were performed and when. This is highly unreliable and introduces a great degree of human error within the dataset. This type of strategy may work for small datasets and experiments, but it is not a suitable solution for most purposes.

- Experience Sampling
  - The idea behind this method is to ask the users periodically and repeatedly which actions they have accomplished. The users must select from a list of appropriate answers. This type of method can be implemented using a mobile application and therefore is suitable for real time labelling. However, the labelling will be limited to the users which are part of the study. Additionally, users may get overwhelmed by the requests and the quality of the label is likely to deteriorate. Furthermore, due to the periodic nature of the labelling certain activities, especially those of short duration, may not be appropriately captured.

- Prompted Labelling
  - Prompted labelling is an approach which relies on directly asking the users about the action they have just performed. Through different techniques, the activities of a user are being monitored and when an activity is detected the user is prompted to describe what action was performed. One of the shortfalls of this method is that it relies on the complete participation of the users trough constant probing. Users tend to lose interest which often leads to the introduction of noise within the dataset. Additionally, the detection of the completion of events does not allow for the distinction of actions that may have been executed in a continuous manner without a pause in between.

- Time Diary
  - Time diary is a method similar to self recall where users are asked to log the activities as they are performing them rather than after the fact. This is typically done using a mobile application. This type of approach is often taken for energy disagreggation labelling. This technique works in real time but is subject to some

limitations based on sampling frequency and user participation. Users must be trusted to enter all the activities that they are performing because down time is also considered for labelling. The complete reliance upon the users tends to introduce a high level of noise due to the low level of motivation of the users who are constantly asked to log their information [40].

- Human Observer
  - This type of approach makes use of human labellers who are tasked to document the actions of subjects interacting with the sensors. This method is expensive in terms of cost as it requires a high number of labellers which in return limits the dataset size. The large scale deployment of such solution for the recognition of various activities is not considered feasible. Furthermore, this solution does not have the potential to be adapted for automated labelling due to its dependence upon third party labellers.
- Audio and Video Recordings
  - Audio recordings can be used for users to document their activities, based on those recordings expert can later apply labels to the data. The main idea behind the use of an audio recording is to mitigate the impact of the labeller on the task being performed. It has been observed that having an observer or video observer present during the accomplishment of various tasks may influence how the task is being performed. Therefore by using speech, we can insure that the label adequately captures the time and duration of the activity without any interferences. However, the cost of this solution is highly prohibitive both in terms of time and computation as it is required to process the speech into labels. This would render this solution infeasible for real time activity labelling.
  - Similarly to audio recordings, video recordings are used for labellers to identify the activity being performed at the time of the reading. Based on the video, annotator can very accurately label the data. Although this solution may be ideal for the labelling of very critical tasks, it is not portable to everyday activities for various sensors due to its high cost and real time impracticability.

2.2.1 Existing Work

Roggen et al. [41] explored the idea of offline and online video labelling. They had observers watch individuals through videos as they completed activities. This solution responds to the need to not interfere with the subjects as they perform activities, in order to limit the introduction of any noise within the data. Although the cost of the experiment is extremely high, it does provide a very accurate labelling. It was noted that the online recognition required high alertness from the observers which could eventually lead to erroneous noisy labelling. The offline solution was considered the real ground truth due to the ability of the observers to review each activity and accurately provide the label without added the stress from the real time requirement.

The work of Harada et al. [13] presents an audio recording approach using speech to label activities as they are being performed. A dual channel approach is used: one component serves as a means to capture the sensor data and the other captures the voice data necessary for the labelling. It has been observed that having an observer or video observer present during the accomplishment of various tasks may influence how the task is being performed. Therefore by using speech, we can ensure that the label adequately captures the time and duration of the activity without any interference; we can increase the accuracy of the label and reduce the noise within the data.

In the work presented by Machado et al. [42] the human observer technique was used. The users were asked to perform tasks while accelerometer data was being gathered. The participants were constantly observed by labellers who duly noted the completion of each task along with the beginning and ending timestamps. This type of approach is expensive in terms of cost as it requires a high number of labellers which in return limits the dataset size. The deployment of such solution for the recognition of various activities is not feasible. Furthermore, this solution does not have the potential to be adapted for automated labelling due to its dependence upon third party labellers.

Cleland et al. [1] presented a prompted labelling approach where their system was constantly attempting to detect the activity performed by the users. Through the use of a mobile application, the system detected when the users ceased to perform an activity and had become still. When such detection occured, the application prompted the user to identify the activity they were just performing. However, although successful within a testing environment it was found that such technique is not very accurate or reliable within real life settings. Multiple activities may take place before a pause is recorded which introduces erroneous and noisy data, making this solution only valid in highly controlled environment.

In an attempt to reduce cost and the number of labels required, Murao and Terada [43] present an approach based on clustering. Users were provided with a list of activities that they should perform in given order. Based upon their memory, the users then performed the tasks. The tasks from each user were then clustered and the label was apposed upon the cluster based on the list of activities and its order. One of the advantages of this approach is that it did not require any additional application or hardware to log the activities. However, such experiments would be required to be repeated on multiple occasions as sensors or activities change. The approach is not easily adaptable to change nor is it suitable for types of sensors that may be continuous and not measuring solely when the activities are being performed.

A multitude of papers focus on modifying algorithm to reduce the cost such as the work presented by Miu et al. [44] who propose strategies to obtain the best annotation results using various annotation services while reducing the cost as much as possible. They came to the conclusion that there exists a maximum number of annotation a user is willing to provide without affecting its labelling accuracy. Additional approaches are proposed to reduce the cost of labelling by modifying the learning algorithms used for analytics. For example active learning methodologies may be used to reduce the number of instances required to train a model [45]. The problem associated with the high cost of data labelling has lead researchers to turn to alternative methods such as unsupervised techniques or active learning algorithms [46]. However, this type of approach still requires a means to provide labelling [47] and at the lowest possible cost.

Furthermore, the quality of the label is directly affected by the performance of the labeller, by their familiarity with the data and with the activities being performed. The labelling obtained through the use of a third party observer is directly affected by the abilities of this observer to remain alert as well as its knowledge of the tasks at hand [17].

In addition to the quality and cost challenges associated with the labelling of sensor data, a third challenge lies within the changing nature of sensor data. Indeed, sensor data streams may continually change and therefore the mining algorithm [48] and labelling methodology should be able to adapt quickly and effectively to changes in order to capture the new data patterns. As new sensors are installed and new activities performed, data should be quickly and effectively labelled in order to provide real time analytics. The IoT is highly dependent upon such real time requirement. The new requirement mposed by the network of connected sensor is that many tasks such as data cleaning and labelling “must now be performed autonomously in real-time”[49].

Therefore, a new means of labelling data is required in order to reduce cost and retain labelling accuracy while easily allowing for real time labelling and easy autonomous addition of new labels. The crowdsourcing paradigm provides such a low cost solution with a potential for real time data acquisition and will therefore be described in the following section.

2.3 Crowdsourcing for Sensor Analytics

Crowdsourcing is a solution that leverages the power of crowds to perform tasks at a low cost [50]. It takes tasks normally performed by dedicated workers and outsources them to a large group of people. In terms of labelling, such task would be to identify and provide annotation for data readings. When it comes to sensor data labelling, crowdsourcing has the potential to provide a solution that responds to the need for flexibility and updatability of the sensor real-time requirement in addition to the cost reduction provided by leveraging the power of crowds rather than the assignment of specialized observers.

2.3.1 Crowdsourcing Challenges

There exists various challenges to crowdsourcing but perhaps the main challenge associated with crowdsourcing is its susceptibility to noisy labels.

By outsourcing the tasks to a number of unspecialized individuals there is a possibility that the data gathered by the worker may become noisy or erroneous due to a variety of factors such as lack of interest or of knowledge by the annotator. There exists various ways to mitigate the noise associated with crowdsourcing.

One approach taken in the academic literature is to create multiple labels coming from various labellers for the same data point [51, 52, 53], a consensus approach can then be taken to attempt to reduce the noise attributed to the labeller and choosing the best label.

Another approach is to eliminate spammer annotators who are defined as poorly performing labellers that are not participating in good faith but rather for ulterior motives such as financial gain. The idea is to perform labelling only based on those considered good annotators [54]. Lastly, another approach is to integrate the concept of soft labelling within the crowdsourcing and sending data with soft labels rather than the typical hard labels [55]. Algorithms can be adapted to respond to soft label and provide interesting accuracy. However, the use of soft labelling is limited and tedious when attempting to create easily analyzable data references.

Therefore, it becomes clear that a strong crowdsourcing platform contains a way to mitigate noisy data labels either through validation of the annotations or through restrictions of the annotators.

Another challenge is related to the real time labelling requirements of sensor data in the context of the IoT. Due to the high volatility and changeability of the sensor data and of its related activities (label), labels should be obtainable in real time in an automated fashion. Therefore, typical crowdsourcing platforms such as Amazon's Mechanical Turk [56] are unsuitable for the real time requirement of sensor data labelling for sensor data analytics. Indeed, synchronous double data channels are often required to merge the labelling and the sensor data [45].

There exists a variety of solutions developed to label sensor data using crowdsourcing which will be discussed in the following section.

2.3.2 Crowdsourcing and CrowdSensing Frameworks

In response to the real time challenges faced by crowdsourcing, a new paradigm variant was developed: crowdsensing. Yang et al. [57] defined crowdsensing as a "new paradigm which takes advantage of the pervasive smartphones to sense, collect, and analyze data". It enables the usage of the sensors built within smartphones to capture and label data according to requested tasks.

Smartphones are considered an integral part of the IoT and play an important role in its deployment, this is often due to the ability of smartphones to act as a bridge between various sensor objects [58] but in the context of crowdsensing they act as information gathering devices themselves.

There exists two main types of crowdsensing: people centric and environment centric. A description of each of these approaches [59] is described below:

- People Centric
  - Integrates smartphones sensors to gather information about the users and the activities that they perform.

- Environmental Centric
  - Uses smartphones sensors to document and label the environment surrounding the users.

MCSENSE [60] is a crowd sensing platform for smart cities, it creates a urban data network based on geo-social data. Based on the users' location they may be asked to perform actions such as taking a picture or a temperature reading. It also matches users based on certain criteria to perform tasks together and rewards its user both monetarily and through social feedbacks. However, it states that infrastructure based sensing is not included in the platform and would require a complementary platform to fully enable smart city sensing.

Various sensor crowdsourcing frameworks are developed for specific types of sensors, each solution either responds to labelling for mobile sensors or for infrastructure based sensors. None of the solutions found in the literature provided a flexible infrastructure that would allow for the real time processing of both mobile and permanent sensors. This lack of adaptability is reflected as a shortfall for real time analytics of the IoT as it is composed of a combination of both sensor types.

The vast majority of crowdsourcing sensor labelling published work utilizes the power of mobile sensors, whether they be health sensors, sensors within the mobile phone or even make use of the users as sensors. However, there exists little work that aim at crowdsourcing the labelling of fixed sensor such as those in smart building powering the IoT.

mCrowd [12] is a crowdsourcing mobile application that brings together three interesting concepts. The use of mobile phone sensors to accomplish labelling tasks though crowdsensing, the mobile access to long running crowdsourcing services such as Mechanical Turk and lastly the integration of the various crowdsourcing services within one application. However, mCrowd does not enable crowdsourcing tasks to interact with external sensors in order to appose labels. Once again, this framework does not fully enable real time data labelling for the entirety of the IoT.

CROSS or CROwdsouring Support system for disaster Surveillance [61] is a platform put in place to bring people together in times of disaster to enable real time temporary replacement of permanent sensors. In time of emergency the coverage of physical sensors may be compromised, therefore the power of crowds is leveraged to bridge the gap and enable the collaboration between physical and mobile sensors. Mobile sensors are leveraged to acquire and label data. Once enough data has been gathered through the crowdsourcing tasks, the process is completed. This framework enables the real time gathering and labelling of sensor data as well as the collaboration between physical and mobile sensors. However, the collaboration is not performed in terms of interactions; there still is not any crowdsourcing tasks through the permanent sensors but rather through complementary tasks. Once again, this framework does not enable sensor labelling for all types of sensors in real time.

These solutions all provide a means of labelling human activities and environmental data but do not quite enable the labelling of infrastructure sensors nor do they enable a cost efficient way of labelling data. However, if users are not motivated to participate in the process, the use of the crowdsourcing paradigm to remediate to the expensive cost of data labelling may not be sufficient to successfully gather hard data labels. A way of motivating users to participate in the labelling process without the financial burden typically associated with crowdsourcing applications such as Mechanical Turk is needed. Monetary rewards are often not enough to ensure quality and continued participation. The solution suggested by this thesis is to use gamification to reduce the labelling cost of sensor data while motivating the users to participate effectively. Gamification concepts and existing solutions will be presented in the following section.

2.4 Gamification

The Gamification Summit defines gamification as the use of game thinking and game mechanics to engage an audience and solve problems [62].

There exists many different techniques to integrate gamification, all of which aim at creating a long lasting and deep engagement between the participants, the non-game activities and the supporting organizations [63].

A key success point of gamification relies on the fact that the gaming feature of an application enhances internal motivations [63]. If one is not interested in the underlying activity, adding external motivation will not be a long term successful endeavour. However if the gaming components can enhance something the individual already cares for, gamification can be a highly successful model.

Gamification requires four main components in order to be deployed [64]. The use of gaming mechanics, a measurement of the users success, a behaviour we are trying to enhance or reduce and lastly a way to reward the user. These components can be implemented in any manner. Examples of gaming mechanics are the use of points, levels and leaderboard. The success of a user may be measured by the number of tasks they have accomplished or the level they have reached. Games are related to intrinsic motivation and must therefore aim at modifying a behaviour or increasing a behaviour, this may simply be to increase the loyalty of a user or to change the users' habits. Lastly, in order for the game to be successful a way of rewarding the users must be integrated. Rewards may be as simple as giving points or providing social feedback through social media.

Through the integration of gamification with basic questionnaire and quizzes, academic research was conducted and the results verified that gamification increases the level of participation [65]. Additionally, it was noted that the gamification process did not negatively affect the quality of the data and that test subject completed the task with similar performance.

Therefore, gamification due to its direct relation with intrinsic motivation is a suitable way to increase and sustain users participation in crowdsourcing tasks. Not only does gamification increase the fun factor of an application, it also enables the achievement of more accurate work, enhanced retention rate and it does so in a more cost effective manner [66].

Additionally, gamification can be combined with big data analytics and provide users with the best possible experience by utilizing gamification as a big data processing engine [67]. This can be achieved by leveraging the power of gamification to consume, label and analyze data.

The following section will present an overview of the literature regarding gamification and its applications in a context of crowdsourcing.

2.4.1 Gamified Crowdsourcing Frameworks

This section aims at providing an insight and discussion on what has been achieved through gamified crowdsourcing in the literature. The findings shall be put into context with what should be achieved in order to enable real time sensor data analytics.

BudBurst [68] is a crowdsensing framework that is designed to capture and label information regarding flora through gaming mechanics, such as, points and levels. The game mimics the idea behind geocaching. The users are responsible for finding and cataloguing plants and flowers. The game serves as a motivation to engage user participation. The results of the experiment showed a heightened enthusiasm in participating. Additionally, the data gathered was used to perform analysis regarding global warming. This framework demonstrates how the integration of gamification with crowdsourcing is highly beneficial to the enhancement of motivation and resulted in better recruitment and retention of the users.

Urbanopoly [14] is a crowdsensing framework that is environmental centric. It is set out as a game similar to monopoly where users compete against each other to acquire and visit locations. The twist being that locations are actual geographical places or businesses. When the users physically visits those places, they get to edit and enter data corresponding to the location such as the name and opening schedule. Crowdsourcing and gamification come together to provide a labelled venue dataset. One of the main issues related to this framework is the tight coupling between the labelling context and the game implementation. Consequently, the gamification architecture cannot be utilized in any other context. Ideally a solution would enable flexibility in the domain of implementation. This is required to facilitate the development of a framework that is flexible enough to support any types of sensors.

Herd It [69] is a crowdsourcing framework that enables the capture of music annotations which it then uses to perform analytics on music data. Furthermore, it uses both supervised and active learning methods to train and request specific data labels. The framework leverages crowds to annotate songs through the use of gamification. Users are placed in groups and presented with a song along with suggested tags that describe the song. Points are granted in relation with the agreements of the tag assignment amongst players of the group. Barrington et al [69] investigate the ability of gamification to not only label data but to power analytics. The findings are that this methodology is as effective and accurate as labelling performed by domain experts.

Similarly, SoundsLike [70] leverages gamification to label movie soundtrack in order to utilize the labels to improve the classification of its content. Although this particular research focus on semantic and audio data, the idea of powering analytics directly from the labelling framework is directly in line with the real time analytical requirement of the IoT. This suggests that using crowdsourcing to power analytics is a feasible and successful approach.

The freemium model [71] is a platform in which various games can be played, however in order to earn power ups to be used within the game users are asked to complete specific tasks. Because the task labelling are deployed across multiple games, this framework ensures a higher production of labels as the appeal of a specific game is not as influential. The game implementation is not related to the actual crowdsourcing task, this level of separation is powerful as the framework may be deployed for any game for which power ups may be applicable. However, the framework is also limited in the tasks it may ask a user to accomplish due to the fact that no sensor or locational requirements have been established. In this particular implementation, users are asked to accomplish motor and cognitive tasks. The limitation of the tasks to be labelled by this framework presents a disadvantage and render the approach inapplicable for real time sensor analytical purposes.

Gamification has been studied and implemented along with crowdsourcing in many different fields of study. It was even implemented to evaluate the performance of a search system in order to gather greater amounts of sample data more quickly, cheaply and effectively [72]. However, of the academic work which was reviewed for this work, none appear to be related with direct intreractions with external sensors.

Additionally, none of the reviewed works provided a means of generating various game applications. Each of the presented frameworks are associated with a specific game or set of games. It does not allow for variance in the way gamification is implemented.

The following section will more clearly define the research gap identified in this thesis.

2.5 Summary

The litterature review presented in this chapter shows the need for a new sensor data labelling technique that would be flexible enough to function on any types of data. With the emergence of the IoT, the new labelling technique must also be adaptable to constantly new and changing sensor data. Current sensor data labelling techniques are too expensive and are not adaptable to this new sensor era.

Additionally, a new means to create quickly and effectively sensor dataset is of utmost importance for data analytics research and the existing crowdsourcing solutions are unable to provide such a solution.

Lastly, gamification has been introduced as a new paradigm to motivate users and is now being used in a variety of contexts. However, a flexible and adaptable architecture to provide means of integrating gamification within a solution has not yet been provided.

The work in this thesis will address these challenges by presenting a flexible gamification framework capable of generating labelled sensor datasets in real time while being capable of adapting to any type of sensors with which humans directly interact.

Chapter 3

Gamification Framework Architecture

The gamification framework described in this thesis presents a solution to the challenges associated with the labelling of sensor data. The approach proposes to leverage the power of crowds by using gamification to capture and label sensor data in order to enable real-time and historical data analysis. By virtue of gamification, we are able to provide real-time labelling of sensor data by associating sensor readings and the physical actions being performed on the entities being measured by the sensors. This responds to the previously described challenges associated to the post-processing of abstract sensor data while promoting user engagement and enabling data analysis. The overview of the framework is shown in Figure 3.1.

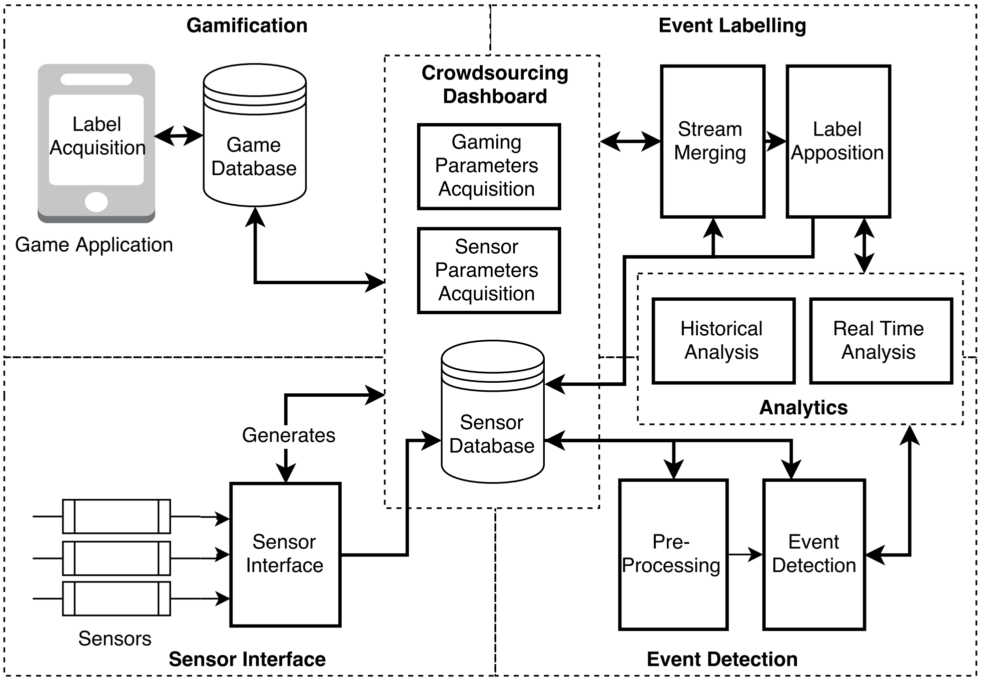


Figure 3.1: Overview of the Framework.

The framework was designed using a modular approach, allowing for flexibility, easy upgradability and maintainability of the system. It is composed of six main modules; crowdsourcing dashboard, gamification, sensor interface, event detection, event labelling and analytics.The idea behind the framework is to use gamification to create gaming related elements in order to accurately label data. Those gaming artifacts within the application will allow for the capture of the activities surrounding sensor readings.

The gamification will enable crowdsourcing by making use of a user bank to accomplish specific gaming tasks. However, the tasks will not be related to the direct identification of the sensor readings but rather related to the identification of what is physically being accomplished during the readings. That is the human activities that are surrounding the sensor readings. The solution will essentially provide a platform for translating sensor data through gamification into human readable labels, these actions will then be used to appose labels on the readings, subsequently enabling data analysis.

The framework relies upon a trust assumption, that is that the majority of the users are being truthful in their participation. Meaning that they are interacting appropriately with the game and are not attempting to cheat. The means by which we detect events and associate labels rely upon this truth model.

There are two main use cases for the framework. The first one is to enable researchers to build substantial datasets by using the framework to crowdsource sensor data. The following example shall illustrate this idea. A researcher may be interested in identifying types of human motion using accelerometer sensors. The researcher wishes to capture accelerometer data to identify whether a user is walking or running. However, it can be difficult and tedious to build a dataset from which to learn. Currently, the researcher could build a small application to collect data on his own, however the size of the dataset would likely be limited to a small number of participants and consequently a small data sample due to the high cost of the data collection.

The framework could be used to generate a much larger dataset quickly and effectively while enabling the researcher to perform real time analysis. The game would be designed to engage the users in physical activities through the promotion of healthy exercising habits. The framework would ask or enable the users to perform gamification tasks, such as “run for 5 minutes” and the data acquired by the sensors would then be labelled appropriately and used to perform analytics.

The second main use case is to employ the framework to obtain data labels in order to perform analysis and extract information from existing sensor data. Indeed, the platform can be utilized to unveil insights and knowledge from data that has been and continues to be gathered by sensors. An example for this use case could be to use the framework for energy disaggregation. Many homes are now equipped with smart meters collecting real-time usage information, this data has now been stored for a number of years but consumers do not readily have access to actionable insights from their data. Implementing the framework in this context would enable the capture of data labels within users' homes and allow for real-time and historical data analysis. The case study presented in Chapter 5 will provide an implementation example for this use case. The framework was designed to function with any sensors measuring data from objects with which users directly interact.

The design of the framework relies on the idea that the users implementing the framework and participating in the data collection are interested and willing to share their data. The privacy aspect related to the unveiling of information was therefore not a concern at this point in the development of this work.

The objectives and the design details of each of the six major components of the framework will be discussed in this chapter.

3.1 Crowdsourcing Dashboard

One of the major contributions of the framework is that it provides a flexible way to gather labels for sensor data. In order to maintain the adaptability of the platform, the framework is designed to work with any sensors capable of measuring data from objects with which users can interact.

The Crowdsourcing Dashboard was designed as the central point of the framework. It is used as the manager which interacts with every other component of the framework and enables the exchange of information necessary to translate the sensor readings and gamification tasks into proper labels.

The key to the translation of the gamification tasks into labelled data reading is the gamification metadata. The metadata essentially describes all the entities necessary to the functioning of the framework, those entities are presented in Table 3.1. The objective of the crowdsourcing dashboard is to provide a means to capture the metadata necessary to the functioning of the framework. The Crowdsourcing Dashboard renders possible the association between the sensor environment and the gaming environment. The main purpose of the dashboard is mainly the acquisition of both sensors and gaming parameters in order to enable that connection.

The crowdsourcing dashboard is essential to the establishment of the framework, it plays a crucial role in enabling the labelling of sensor data through gamification tasks. The entirety of the information required by each of the component to perform their functions is held within the *game_backbone* entity which is responsible for holding the information required to integrate gamification.

The dashboard is a means of entry for all of the entity required by the framework. In order to create the communication layer between each of the components of the framework, specific parameters are required from the users. The communication parameters required can be separated into two sets: the sensor parameters and the gaming parameters. The functionalities related to the acquisition of those parameters will be presented in the subsequent subsections.

3.1.1 Sensor Parameter Acquisition

The translation between the sensor data and the labelling through gamification relies on the sensors contextual information and requirements. This information is held within the framework's metadata which is a combination of a number of entities. The sensor related components of the framework are designed using the entities and relationships presented in Figure 3.2.

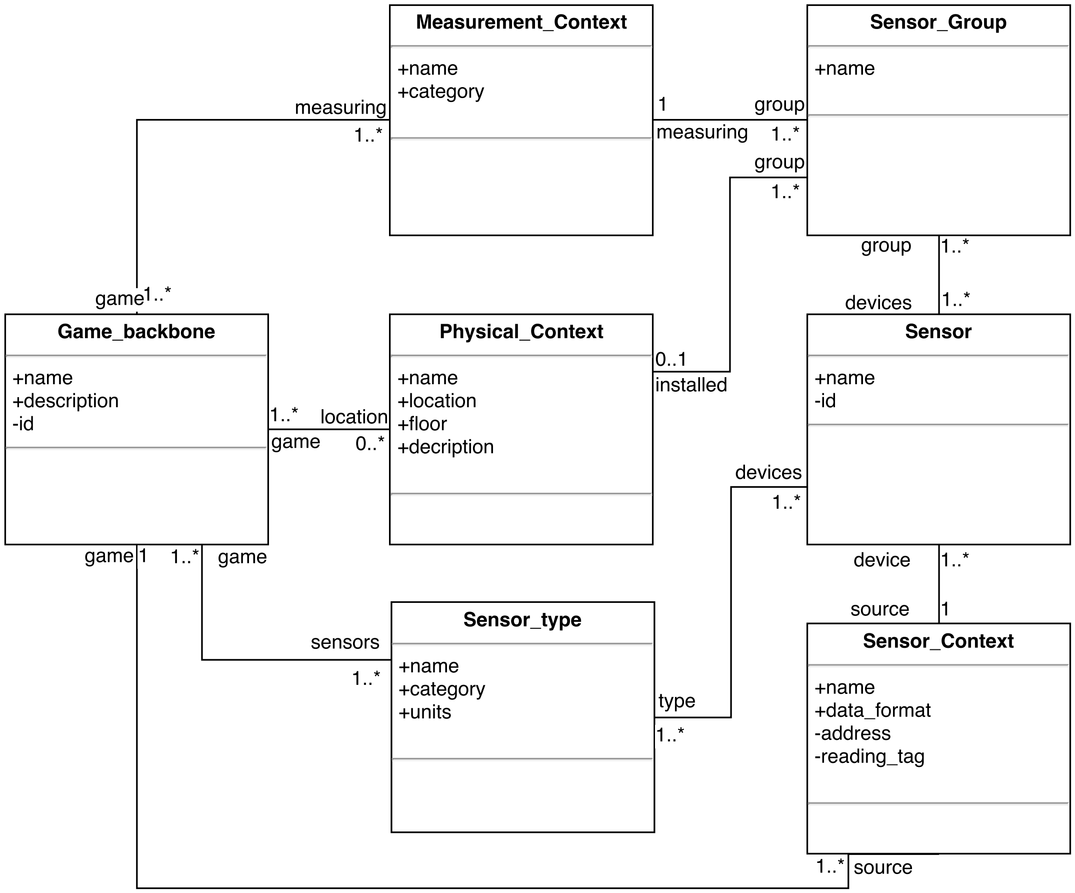


Figure 3.2: Crowdsourcing Sensor Entity Diagram.

The *game_backbone* entity represents the metadata holder of the framework and is therefore a part of various components of the framework. The different modules can use this entity to gain access to all of the information they require within the framework.

A multitude of information is required for the proper deployment of the framework in order to accurately transform gaming events into sensor labels. This includes the details required in order to obtain the sensor data and extract the sensor readings or the contextual information expressing what the sensors are actually measuring. This type of information is held in the various context artifacts shown in Figure 3.2.

Table 3.1 provides a description of each of the entities.

Game_backbone	This entity is an important object of the framework. It creates the bridge between the sensors and the physical actions. It holds all the sensor requirements, context information and gaming parameters.
Physical_Context	This is the virtual representation of where the different sensors are physically installed, this represents where the users are interacting with the devices. It is the physical requirements of a game. This context is optional when the sensors are not tied to a specific location.
Measurement_Context	This represents the context of sensor measurements. This includes information such as measurement categories and types. An example would be HVAC and lighting data.
Sensor_Context	This entity is used to represent the parameters required to translate the sensor data into the data reading entity. It enables the sensor interface to appropriately map spatial temporal data obtained from the sensor to the data reading entity used by the framework.
Sensor_Type	This represents the actual sensor type that is required by a game. This could be things such as voltage sensors and accelerometer. It shall also characterize the sampling rate.
Sensor	This represents the actual sensor device that is gathering data.
Sensor_Group	This represents a group of sensors measuring the same objects within the same physical context.

Table 3.1: Crowdsourcing Sensor Entity Defintions.

It can be observed that the metadata, represented by the *game_backbone* entity, is required to establish the translation between gaming events and the sensor readings. This idea also serves as the central contribution of the gamification component of our work. It outlines how to essentially connect two context using gamification.

The metadata is compulsory to the design of all of the other components of the platform as they utilize this metadata to create and capture the various sensors and gamification artifacts. This includes the set of physical and contextual requirements of the sensors as well as the type of action measured by the sensors.

Once the sensor metadata is entered, the gaming parameters are required in order to complete the connection. The following section will describe the acquisition of the gaming parameters.

3.1.2 Gaming Parameters Acquisition

The gaming parameters enable the abstraction between tangible actions and data readings. The crowdsourcing dashboard serves as a means to setup all of the gaming metadata used throughout the entire framework. The game metadata shall contain all possible targets and actions the framework needs to monitor or encourage. Additionally, a gaming reward artifact shall be associated with each target. This will enable the relationships between targets and gamification components. Figure 3.3 depicts the design of the gamification related metadata, held by the *game_backbone* entity.

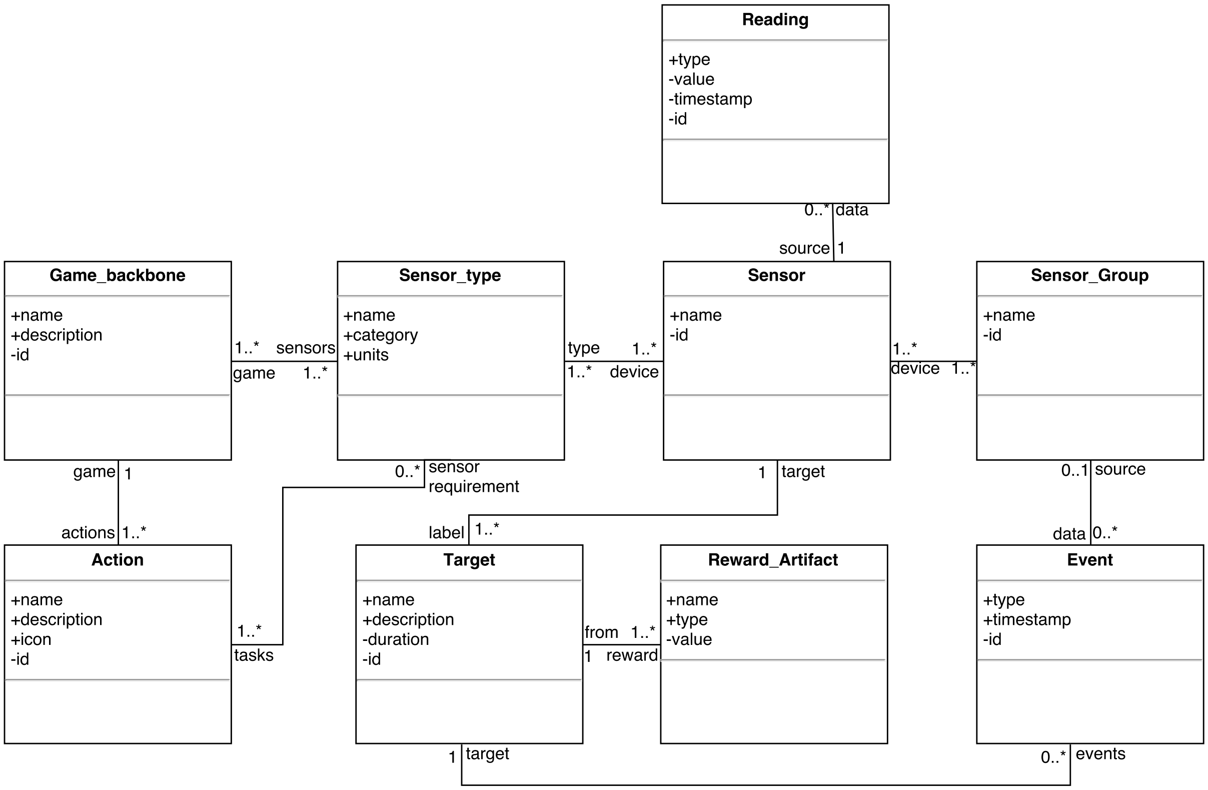


Figure 3.3: Crowdsourcing Gaming Entity Diagram.

The *game_backbone* entity holds the actions and targets associated to the various sensors. The sensor and sensor_type entities are also present in this diagram as they play an essential role in linking the components. Details regarding the other entities can be seen in Table 3.2

Game_backbone	This entity is an important object of the framework. It creates the bridge between the sensors and the physical actions. It holds all the sensor requirements, context information and gaming parameters.
Reading	This is the actual reading coming from a sensor.
Action	Those are the actions to be physically performed by the game users.
Target	This represents the specific details of the action. This will enable us to label the data readings.
Sensor_Type	This represents the actual sensor type that is required by a game. This could be things such as voltage sensors and accelerometer. It shall also characterize the sampling rate.
Sensor	This represents the actual sensor device that is gathering data.
Event	This is the sensor representation of an action being taken on the physical environment.
Reward_Artifact	This is the reward granted for the completion of a target. By enabling custom rewards, various targets can be encouraged in a more controlled manner

Table 3.2: Crowdsourcing Gaming Entity Definitions.

The metadata serves as a structured vessel to hold the data necessary to integrate gamification within the framework. The various artifacts will be implemented by various classes during the implementation stages. Most of this information will be stored in the sensor database. The design of this component will be described in the following subsection.

3.1.3 Sensor Database

As shown in Figure 3.1, the sensor database shall serve as the central data holding component of the framework. As events are detected, they shall be sent to the sensor database. Simultaneously, the labelling process is also taking place and the labelled data shall also be saved in the database, linking the detected event with its appropriate label.

The analytical component also relies upon the sensor database in two ways. Firstly it shall retrieve the previously labelled data to perform its analytical duties. Secondly, it shall also send the results of its analysis back to the database.

Furthermore, the database is designed to hold all the required artifacts such as locational information, sensor types, sensor groups, measurement context, etc. However, the design of the database is highly tied to its implementation and can therefore not be fully detailed.

The sensor database is designed as a central point of entry for all of the sensor data. It shall receive data reading objects from the sensor interface. The sensor database requires sensor readings to follow the format described in Figure 3.4

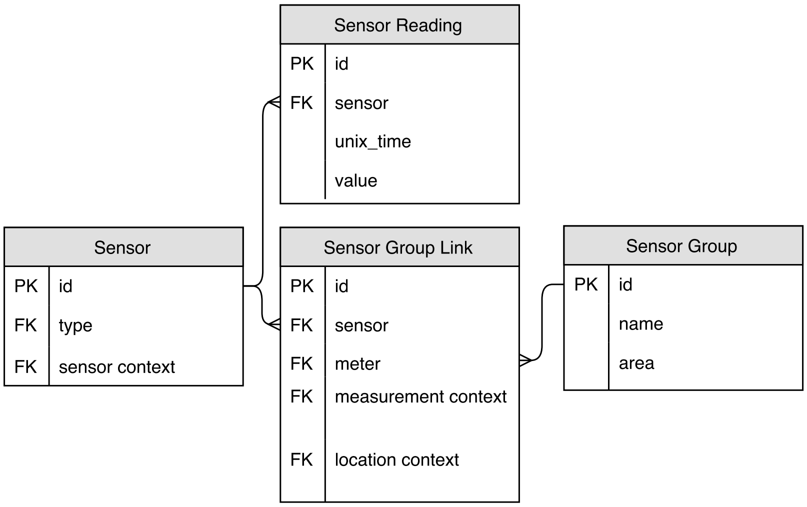


Figure 3.4: Sensor Reading Database Design.

The sensor interface which enables the framework to connect to any types of sensor and send the readings in the proper format will be described in the following section.

3.2 Sensor Interface

The success of the framework depends on the ability of the sensor interface to adapt to any sensor data format. Many sensors provide the users with some type of API to obtain the sensor readings. However, there does not exist a standard in terms of data format for those readings.

Therefore, a flexible and maintainable solution is required to interface between the sensors and the rest of the framework. Although different in their content and format, sensor readings typically provide spatial time series data. By definition, this type of data contains: attributes, object, time and location information. Based on this insight regarding sensor data, the role of the sensor interface is to utilize the sensor_context of the metadata to adequately map the information coming from the sensor stream.

Figure 3.5 depicts the required sensor interface design.

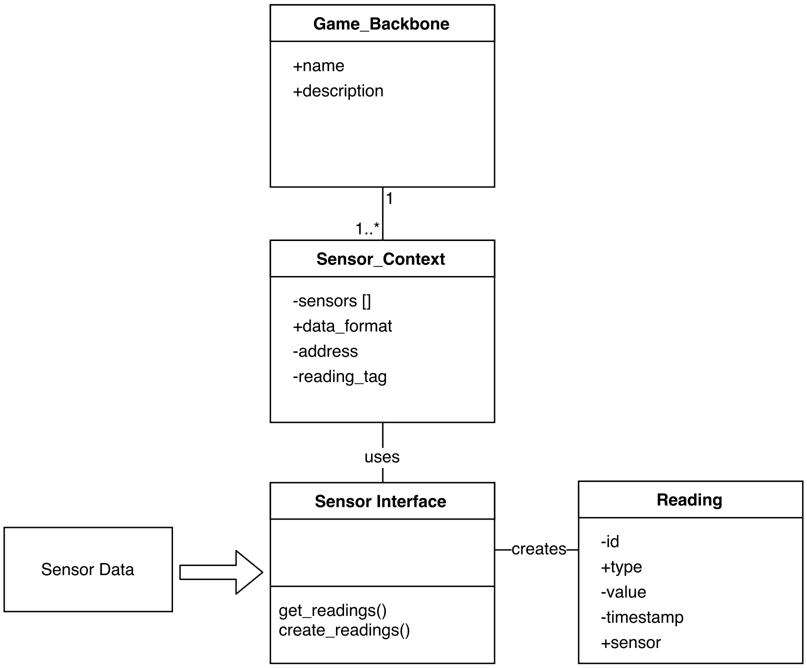


Figure 3.5: Sensor Interface Design.

Once, the sensor data is acquired a means of capturing actions through gamification is required, this will be discussed in the following section.

3.3 Gamification

Gamification is the process of integrating game mechanics in non-game context in order to solve problems. It has a proven impact on user retention and continued user participation when implemented successfully. It is an innovative and entertaining means to motivate users to participate in different programs. The idea behind its application in this framework is to motivate the user to perform the targeted actions through the use of gaming mechanics. The completion of those actions is what will enable the labelling of the data. As users are physically accomplishing actions, the sensor readings will simultaneously be associated, therefore providing the required data labels necessary for data analytics. Gamification enables us to acquire labels in a very inexpensive manner both in financial and computational terms. The game application implements the gamification aspect of the framework. It is responsible to motivate and engage the users through the accomplishment of tasks and missions. Its design will be discussed in the following subsection.

3.3.1 Game Application

The objective of the game application is to enable the communication of the accomplishment of actions from the users to the dashboard. The intent is to use the sense of accomplishment and positive reward system provided by the gamification system to retain the users participation in the label gathering system. Each of the four key components of gamification are embedded in the design of the application in a way that enables customization.

- **Mechanics**: the building blocks of a game such as missions, leaderboards and objectives.
- **Measurements**: the metrics used to quantify users progress.
- **Behaviour**: the behaviour the game is attempting to promote.
- **Reward**: the compensation given to a user for participating.

The game application is designed to easily interact and reflect changes and choices made by administrators to ensure the maintainability, adaptability, and portability of the platform.

In order to incorporate the gamification mechanics, the gaming components needed to be integrated within the existing entity hierarchy as described in the crowdsourcing dashboard section 3.1. The gamification entities will use the *game_backbone* metadata to extract the required information.

Figure 3.6 shows the relationships amongst the objects. Each of the gaming artifacts can be extended and implemented to meet any desired objective. For example, Figure 3.6 shows two classes of actions that inherit from the action class: the scanning_action and timing_action classes. These entities demonstrate how the action object may take multiple forms within the game application.

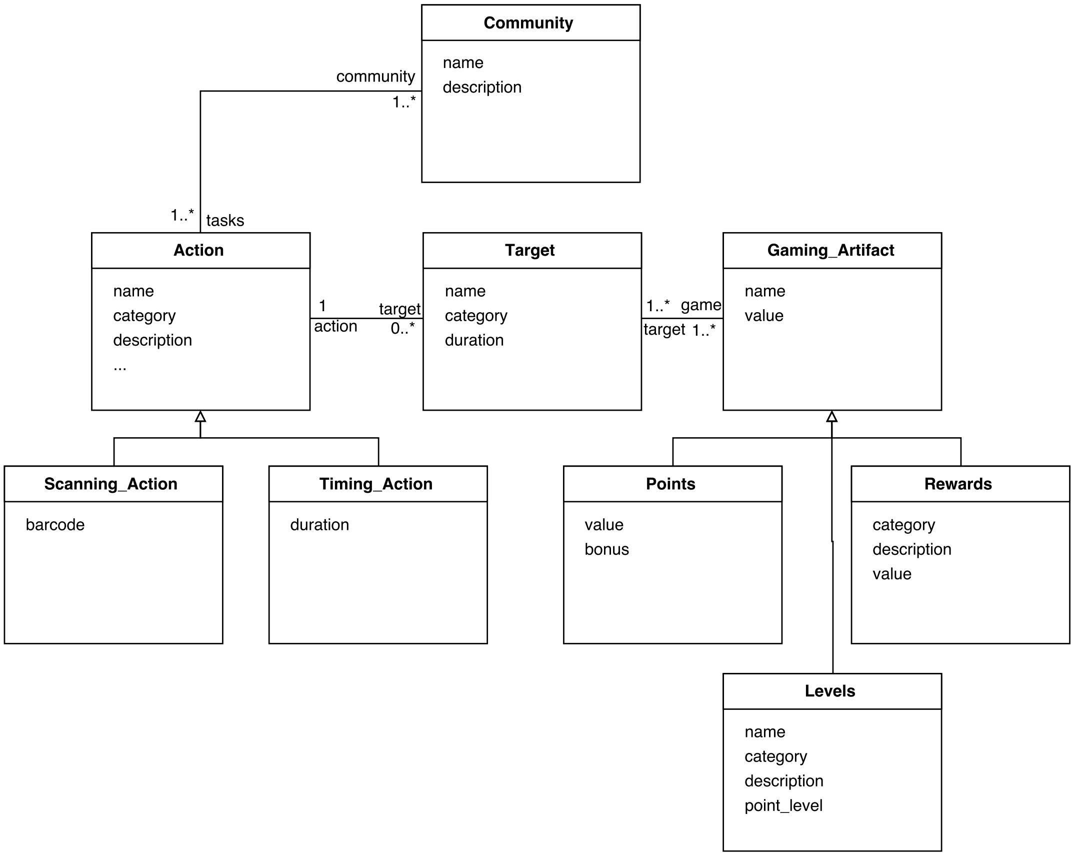


Figure 3.6: Gamification Object Relationships.

The behaviour artifact represents the theme of the actions within the *game_backbone* of the game application. This artifact represents the background theme we are trying to encourage or discourage. Figure 3.7 provides example of classes that may implement this artifact. By explicitly representing the behaviour artifact, linkage, and analytics is made possible across various communities.

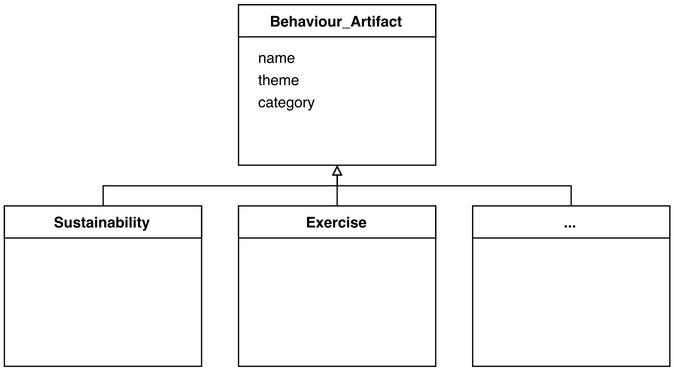


Figure 3.7: Behaviour Artifact Classes.

The game is designed in such a way that the actions shall motivate a change in behaviour, whether it be to start performing more actions of a certain type or to make better choices in the actions taken. As shown in Figure 3.7, examples may be to ask the user to perform sustainable actions in order to track utility sensors or ask the user to increase their physical activity and promote fitness in order to track health sensors.

Reward artifacts are associated with each task, which will enable us to measure the success of the user. Each time a task is accomplished, the user is granted the associated gaming reward. Rewards may be implemented in various ways, examples of such classes are presented in Figure 3.8.

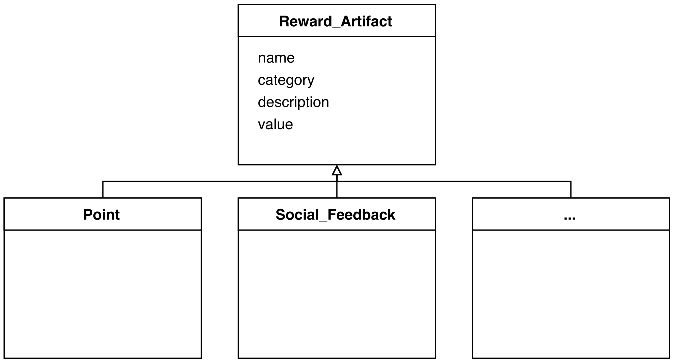


Figure 3.8: Reward Artifact Classes.

The integration of gamification is driven by the incorporation of gaming mechanics within the application. Those mechanics are represented in this design by the mechanic artifacts. Each game can choose to implement various components. Those components interact with the rewards and those rewards are made available and organized by the mechanics. The mechanics are the building blocks of gamification. Examples of such classes are presented in Figure 3.9.

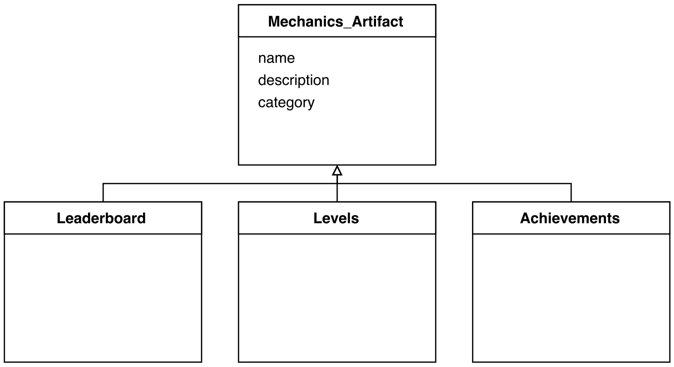


Figure 3.9: Mechanics Artifact Classes.

Since the idea behind using gamification is to facilitate and encourage the interaction between sensors and users, the game application must provide a means to acquire and capture the accomplishment of actions. The following section will outline this part of the design.

3.3.2 Label Acquisition

The game application must allow for the acquisition of gaming events. However, the means of capturing those events in the most optimal way will vary based on the different requirements of the games. The idea behind the label acquisition is to enable the link between the accomplishment of an action and its specific target. The label acquisition module must therefore be flexible in its design as it may be implemented in various different ways based on the diverse requirements of the sensors.

The label acquisition must simply enable the user to select the appropriate target for the action they have chosen to perform. It must allow the user to signal the beginning and/or the end of the action they are performing. In order to maintain a flexible design, the application was designed to rely upon an action_capture interface which will enable the linkage between actions and targets within each game. The action_capture is specific to the type of action selected by the user, based on what they select the appropriate capturing interface shall be used. The label acquisition process is shown in the sequence diagram shown in Figure 3.10. It shows the flow of execution within the application when a gaming task is accomplished by a user.

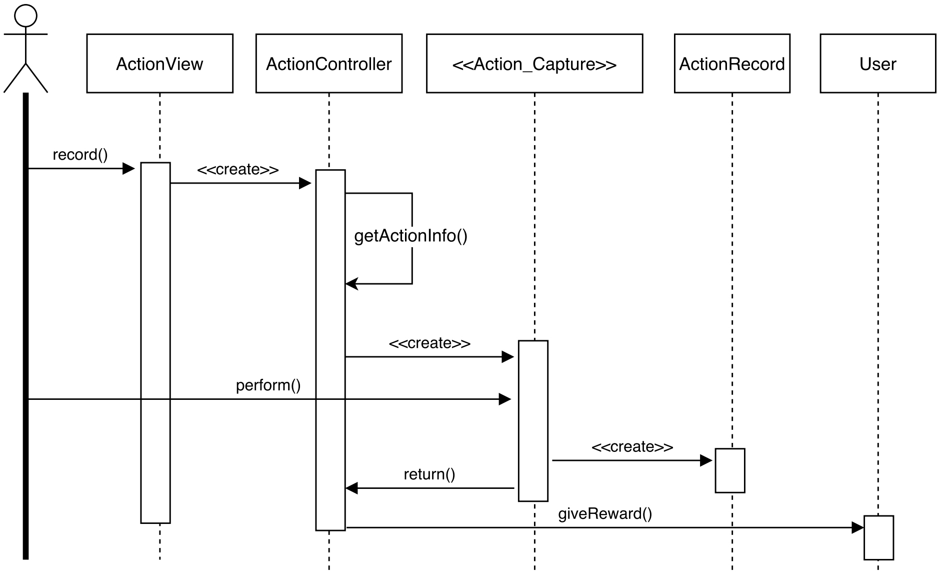


Figure 3.10: Action Record Sequence Diagram.

Figure 3.10 presents the action capture interface. This class will enable the game to be design around any type of capture methodology, this could be through scanning a barcode, or starting a timer for example. Figure 3.11 shows the action capture interface.

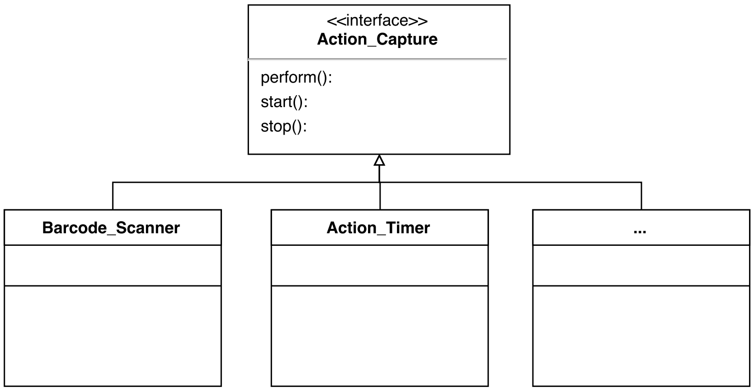


Figure 3.11: Action Capture Interface.

The game application also includes different type of actions, known as missions, the idea behind this concept will be presented next.

Mission Generation

One of the keys to the success of gamification relies upon the user motivation. It is critical for the user to feel motivated and engaged in order to retain participation. Additionally, the statistical distribution of events upon which the event detection and classification algorithm depends is critical to the functioning of the framework. However, due to the human factor involved in the user picking and choosing which actions to perform, we cannot guarantee that all the actions will be performed equally or performed at all. Therefore a mechanism is necessary to ensure that all actions are performed and follow a normal distribution. For this purpose, the idea of mission was introduced.

Periodically, at an interval set during implementation, the number of acquired labels per specified target for each sensor will be monitored by the crowdsourcing framework. The distribution of each target will be calculated and evaluated against each of the target of the sensors. If it is found that certain targets either have a much lower number of labelled instances, the crowdsourcing will generate a mission for the appropriate target.

A mission can be defined as a gaming task, however it differs from regular actions by the fact that it is to be executed on a specific target. Mission objects are generated automatically from the crowdsourcing dashboard and sent to the game application through the game database.

Missions have been designed to be much more interactive than actions in the sense that the user should be notified that a new mission is available. In order to make the missions more exciting and rewarding, the missions should possess a much higher reward value than regular actions, they should also only be available for a limited time. Once a mission has been completed for the specified number of times or when its lifetime expires, the users will no longer be able to complete it.

Once a user selects the mission, the mission is accomplished using the same method previously described for actions.

The game application design is dependent upon the design of the game database which will be introduced in the following section.

3.3.3 Game Database

The game database upon which the game application relies, was designed to be completely separated from the sensor database and the rest of the framework. This degree of separation was necessary in order to ensure the portability and flexibility of the framework and in order to ensure that any changes made to the actions metadata could be immediately reflected in the game. This separation enables the addition of new data labels through the creation of new actions and/or targets in real-time.

The actions and targets created and selected within the crowdsourcing dashboard are created and then sent to the game database. The game application has access to the actions available to the users based on pre established but updatable *game_backbone* metadata. The actions are then shown to the users along with the reward associated with completing the actions. Figure 3.12 shows the process required for the game application to obtain the available actions.

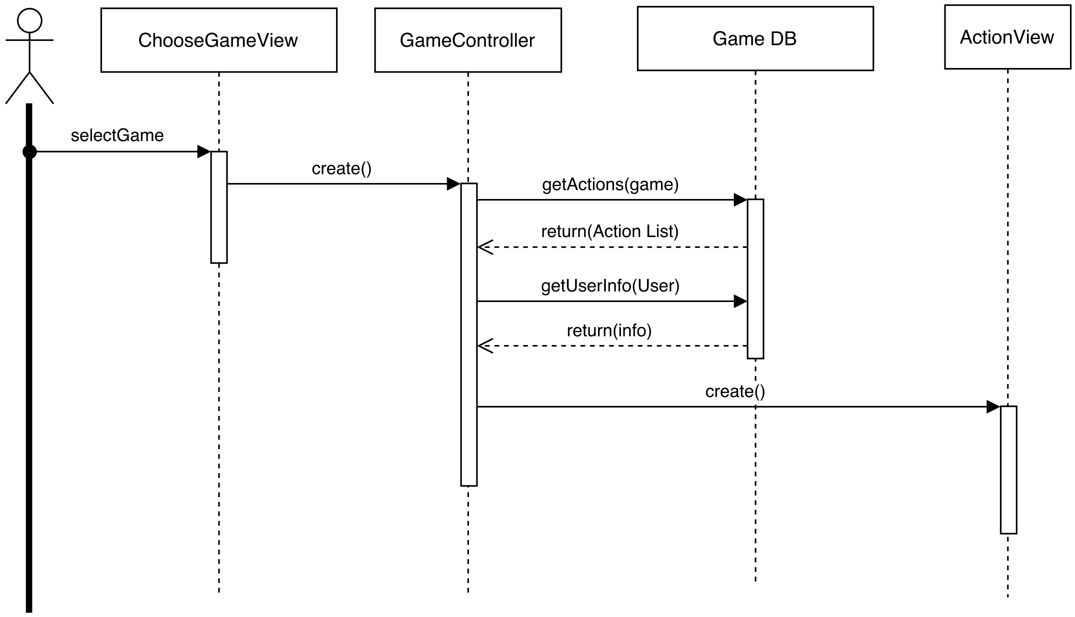


Figure 3.12: Action Polling Sequence Diagram.

By keeping this database separated from the sensor database we are insuring a level of independence amongst all the components of the framework. By doing so, we are allowing for the framework to be deployed across multiple locations and over multiple instances without compromising the accessibility of the platform to the users. Administrators may choose to independently deploy the server side of the framework within their own infrastructure, therefore keeping complete control over their sensor data while still being able to make use of mobile applications and easily enable access to the application to their users.

Conversely, other administrators may wish to share their data with other facilities in order to gain even more insight over their data. By sharing a sensor database across various facilities or sensor banks, learned labels can also be shared, therefore enabling a wider variety of analytics. Regardless of the chosen implementation of the framework, by keeping the gaming component centralized and separated we are still able to leverage the full power of crowdsourcing.

Figure 3.13 shows how the framework may support various distributions of the server side as described above. It can be seen on Figure 3.13 that the game database remains central regardless of the sensor configuration chosen by the administrators.

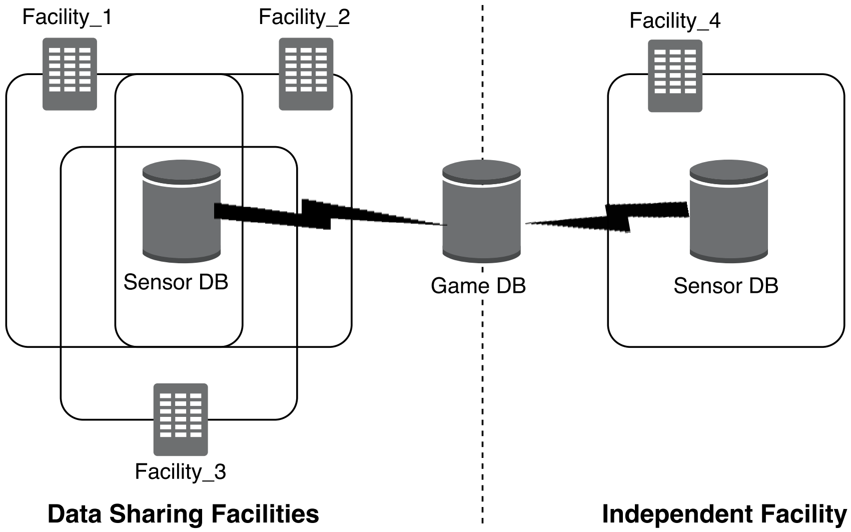


Figure 3.13: Game Central Architecture.

Additionally, Figure 3.13 shows how sensor databases may be shared across facilities. Facilities can therefore also be seen as sensor banks.

3.4 Event Detection

In the context of this framework an event is defined by the readings associated with a change in measurement due to the occurrence of an action. In order to provide accurate labelling, the framework is highly dependent on its capability to detect or recognize that an event was recorded by the sensors.

The idea of solely linking data readings and labels based on timestamp was explored but was not found to be sustainable for many reasons. Firstly, there may be delays between the performed action and the sensor reading, meaning that the timestamp may not coincide. Additionally, depending upon the granularity of the data many readings may be received with similar timestamps. A more robust technique is required to appropriately detect events and then associate the labels. We propose to monitor sensor events separately from gaming events and to then link corresponding events based on a number of rules.

The purpose of the framework is to provide labelling in order to perform data analysis, it is therefore important to identify events in order to extract knowledge from the data. Additionally, in order to provide support for real time analysis, it must be able to detect the events as soon as they are occurring.

3.4.1 Pre-processing

In order to quickly and effectively detect events and respond to the real time requirement of the framework, the event detection algorithm must remain accurate while performing quickly. A number of data analysis event detection techniques are executed in post processing and therefore can allow for an algorithm that is more expensive in terms of time and computations. The algorithm required in this framework must be capable of detecting the events in real time and on large quantities of data.

Given the real time nature of the framework and the high sampling frequency of sensors we can assume that overall the vast majority of sensor readings are not recording an action being performed. Therefore, we can explore the idea of converting normal readings into anomalous ones. The algorithm presented in this section explores that idea through the use of contrast extraction as a pre-processing step. Indeed, we can expose the anomalous properties of events by showing how the readings differ from others.

Therefore by analyzing and recording the changes between the current and previous readings we are extracting the contrast between each reading. We can then apply regular anomaly recognition techniques to our data in order to recognize the events. Equation 3.1 describes the contrast extraction pre-processing step.

$$\Delta_x = \alpha_{x1} - \alpha_{x2} \tag{3.1}$$

$$\forall \text{ sensors x}$$

Where:

α_{x1} represents the previous reading of sensor x and

α_{x2} represents the current reading of sensor x

This pre-processing step can be thought of as a way of exposing the changes between each reading, we can then proceed to monitor those changes in order to establish when a change is representative of an event. The changes are then being monitored for each sensor independently.

3.4.2 Event Detection

When an action is performed it can often relate to multiple sensors, for example when a light is turned off, the kilowatts and the voltage may all vary. Moreover, there may be multiple sensors measuring each of these features and each feature may also vary differently for each target. Therefore, in order to detect and identify a specific event we should be looking for variations of multiple variables across different sensors. However, performing real time monitoring and complex calculation over multiple sensors may greatly impair the real time performance of the framework.

It may be sufficient to detect that one of the features has varied in order to deem that an event has occurred. This idea proposes that if we could leverage the power of univariate detection for multivariate events, the processing time may be improved. Additionally, in order to successfully label the gaming event, we are more concerned in catching all possible events with some false positive rather than missing some potential events. Therefore a slightly less accurate but faster technique is desirable.

Because of the unknown nature of each of the sensors, no assumptions can be made regarding which sensors may vary together without having intimate knowledge and understanding of what is being measured. Each sensor is associated with a measurement_context (electrical lighting, hvac, etc) and it can be established that all the sensors of the same measurement and physical context are varying together. Therefore, all sensors belonging to a same object or sensor group shall require to be linked for the labelling of the event.

For each sensor we monitor the rolling mean of its contrast, if a reading variance is greater than the expected measure, the reading is found to be anomalous and an event is deemed to have occured. Following this detection, all readings in provenance of other related sensors are also grabbed and the detection is terminated for this sampling of readings. This enables us to parallelize the detection and improve the real time performance of the algorithm which is presented in Figure 3.14

```
Data: Sensor Readings
Result: Event Data
initialization;
foreach sensor of group m do
    while Event not detected do
        read current data;
        check sensor rolling mean;
        if detects then
            get all current readings from m;
            insert the current readings into an eventReading;
        else
            get next sensor reading;
        end
    end
end
```

Figure 3.14: Event Detection Algorithm.

3.5 Event Labelling

The core of the framework is based upon its ability to label events, through this process we will gain the labelled data necessary for data analysis. The idea behind the event labelling process is to correctly join the gaming event with its corresponding sensor event. While the sensors are capturing and monitoring readings looking for events, users are concurrently performing gaming actions. As the actions are being performed, the labels of the actions are being captured by users with the mobile applications.

The event labelling process depends on the framework's ability to merge the appropriate streams and to correctly pair events with their labels. Both of these processes, the stream merging and the label apposition, will be explained in the following subsections.

Figure 3.15 shows how the framework is built around two separate data streams: the gaming data and the sensor data streams. The event detection module is designed to catch all possible events with an emphasis on not missing any events, even if that includes detecting a few false positives. The false positives are then dealt with during the label apposition process which is required to be much more strict then the event detection process in order to ensure the veracity of the labelled data outputted by the framework. as shown in Figure 3.15

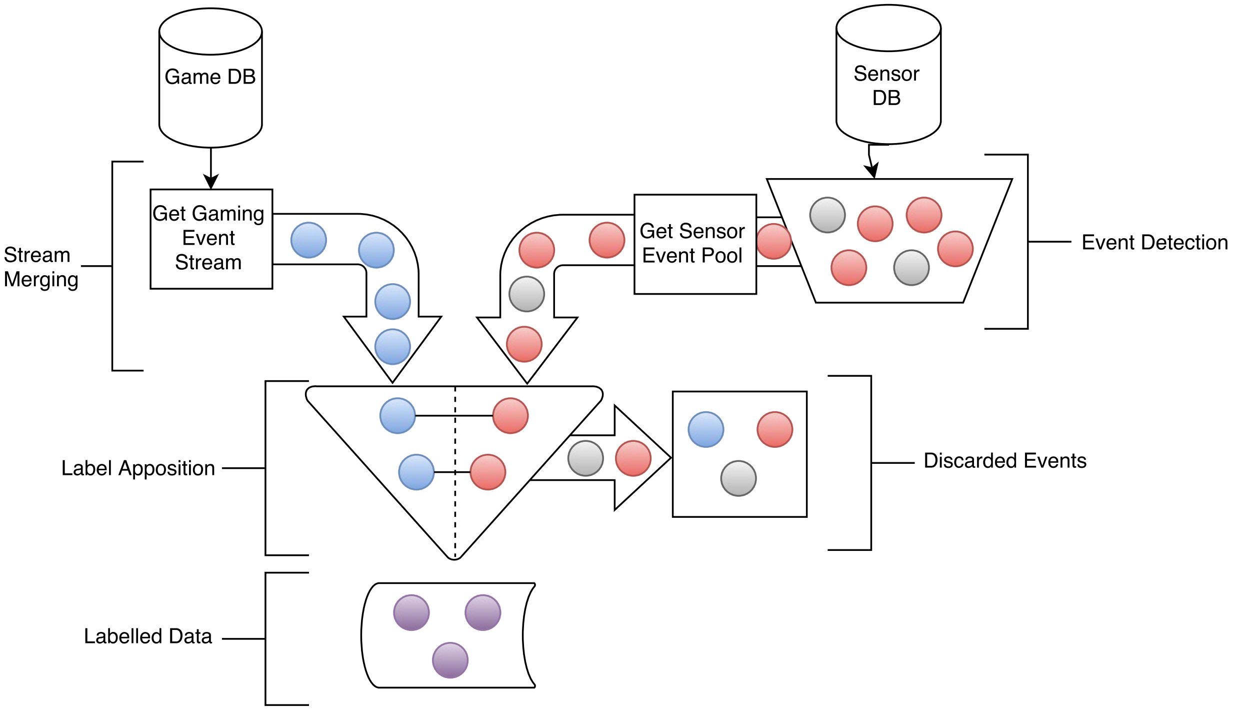


Figure 3.15: Label Apposition Process.

3.5.1 Stream Merging

Based upon the contextual information associated with the performed actions, we are capable of merging the detected sensor readings and the gaming events. The crowdsourcing dashboard is responsible for polling the game database for all the gaming events of its sensor targets. The event labelling module then separates the events in terms of sensors based on the targets of the gaming events. This effectively creates the gaming event streams for each of the sensors. The gaming event streams are then brought together with the appropriate sensor event pool and the label apposition process may begin.

3.5.2 Label Apposition

Given the gaming nature of the framework, not every user input can be taken as true. Considerations must be given to the users actions. Therefore we must be careful in the label apposition and must take the opposite approach than that of event detection. We would rather miss an event than have a false positive because one of the goals of the framework is to create a hard label dataset with little noise.

It is critical for the framework to discard any sensor or gaming event in which it is not confident, otherwise the validity of the framework would be highly compromised and noise would be introduced in our dataset. The main purpose of this solution is to enable the creation of sensor datasets in real time. If labels are incorrectly apposed, the framework would no longer be serving its purpose as the analytical consequences would be significant. The results of the analytics are directly influenced by the labels we acquire.

In order to ensure that only hard labels are obtained in our labelled dataset, a firm set of rules is required. The apposition occurs in three steps:

- Only assign labels to event if they fall within a strict sliding window.

- Only assign labels to multi-events if a corresponding label can be found for the combination of these events.

- Once labels are assigned, perform *k*-means clustering using all the data from a specified target. If the new reading belongs to the same cluster has the remaining labelled readings, keep the newly labelled reading. Otherwise, discard the data reading.

The algorithm presented in Figure 3.16 is composed of the firm set of rules presented above in order to ensure that the validity is preserved. In this case, discarding events is much more desirable than making a match in which we do not have total confidence.

The inputs of the algorithm are: the event data detected from the sensor stream, the gaming data and a sliding window corresponding to the sampling rate of the sensor data. The sliding window is important in order to appropriately match the timestamps.

```
Data: Event Data, Gaming Data, sliding window
Result: Event Data Labelled
initialization;
while Gaming Data is Available do
    get gaming data;
    get appropriate event data stream;
    if event was detected then
        check if single event is found in sliding window;
        if one event then
            Associate event with reading;
            Perform clustering to ensure label is accurate;
        else
            if multiple events then
                Assess the reading similarity;
                if close similarity then
                    associate label
                else
                    discard the event
                end
            else
                discard the event
            end
        end
    else
        get next sensor reading;
    end
end
```

Figure 3.16: Label Association Algorithm.

Once the labels have been apposed, data analytics is enabled. The design of the analytics component is presented in the following section.

3.6 Analytics

The framework was designed to enable different types and levels of data analytics. The platform is designed in such a way that the sensor data can be shared across different distributions.

Depending on the chosen implementation of the framework analytics may be performed using either shared or local sensor data.

A shared distribution requires to share measurement contexts and sensor types along with the sensor readings. By making use of those known context, known labels may be used across facilities to perform analytics. This enables more in depth analysis for similar sensor groupings even if each of the individual facilities do not possess large amounts of sensor data individually. Figure 3.17 depicts the relationship between algorithms and the sensors on a standalone version whereas Figure 3.18 shows how algorithm can also depend on measurement context and sensor type which enable analysis on a shared set.

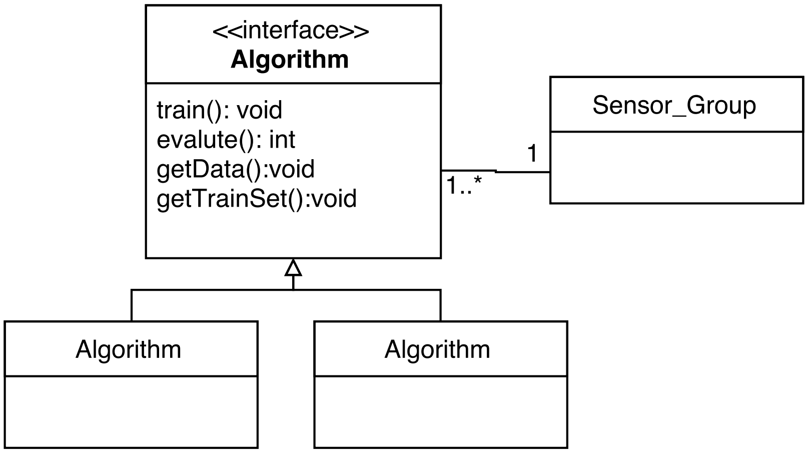


Figure 3.17: Standalone Analytics Classes.

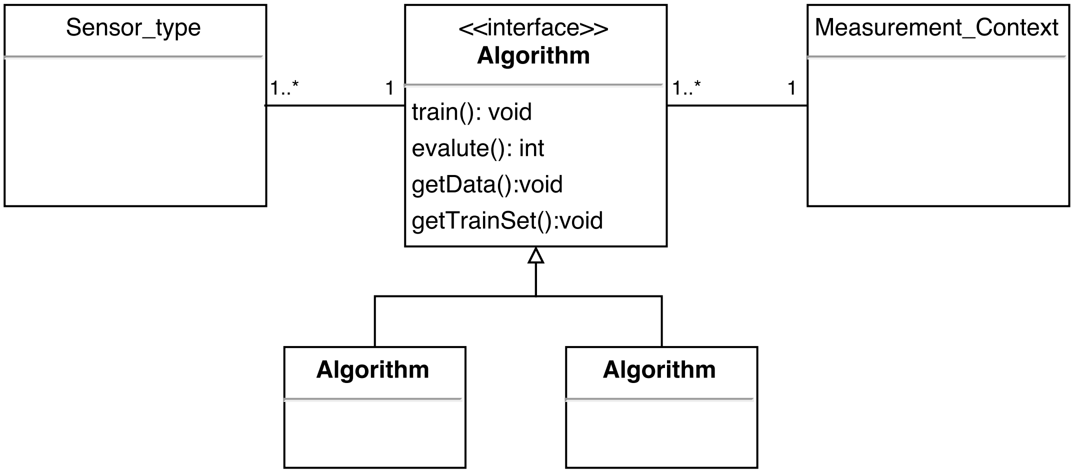


Figure 3.18: Shared Analytics Classes.

This enables different administrators to gain insight on their data more quickly and effectively. Additionally, data analytics can be performed historically or in real time. The data label acquisition is at the centre of the data analytics capabilities of the framework because it provides researchers with labelled training sets. The platform enables real time analytics of sensor data based on the information provided by the system administrator. Due to the real time event recognition of the framework, both real time feedback and historical knowledge extraction can be performed. The following subsections will describe how each type of analytics can be performed.

3.6.1 Real Time Analysis

The framework was designed to recognize potential events in real time. Infrastructures were put in place to extract those events and as the game progresses, more and more of those events will be labelled. This enables the framework to actively learn from those labels. As each of the events are recognized, the framework is capable of performing real time classification of the events based on the previously acquired labels. As more labels are gathered, the more accurate the real time analysis will become.

The real time analysis component is not tied to any specific classification algorithm or analysis implementation. For each of the sensor group, real time information can be extracted, based on the targets entered by the administrator.

3.6.2 Historical Analysis

Based upon the event classification further analysis and knowledge extraction is available to the users. Furthermore, in addition to real time analysis, historical analysis can be performed within the framework in order to apply the newly found knowledge to extract additional information from historical sensor data.

3.7 Summary

This chapter introduced the architecture of the gamification framework presented in this thesis. Through the design of its six major components various contributions were made.

First of all, a novel data labelling methodology for sensor data was shown. Indeed, by means of gamification, the data labelling technique was made flexible, adaptable and inexpensive while enabling easy addition of new data labels. Through its architecture the technique is capable of responding to the real-time data labelling and fast adaptation requirement that emerged from the IoT [73]. Indeed, the IoT is built on and for real-time information, which is required for devices to communicate and responds to various changes. Our framework, allows for quick deployment of real-time data labelling for any new or existing sensors, enabling the intelligent decision making and knowledge extraction at the core of the IoT. Furthermore, the data labelling process is rigorous yet computationally inexpensive in order to address the need for the real time obtention of datasets with low noise level.

Secondly, through its flexible sensor interface, the framework is designed to function with any types of sensor, whether mobile or permanently installed. The crowdsourcing dashboard enables the easy and effective mapping between sensor data sources and the framework.

Additionally, the crowdsourcing framework in combination with the data labelling module, the gamification and the sensor interface provides the required components to serve as a sensor data crowdsourcing service. The crowdsourcing framework allows for easy data task requests and the game application is designed to make the tasks accessible to a large number of participants. Based upon the various contextual requirements, users can join games they wish to participate in and enable large scale data labelling acquisition.

Furthermore, the *game_backbone* entity is used to relate the various sensor contextual information to the gamification artifacts. Through the use of this metadata, gamification can be deployed in any context. More specifically, the use of the metadata enables a complete separation between the sensor data component and the gaming label acquisition component. The framework can be implemented for any types of game, using any combination of gaming mechanics, rewards, measurements or behaviours. As long as a game is designed using those components, the framework will be fully functional.

Lastly, the entire framework allows for real time and historical data analytics by leveraging its event detection component and the labelled data it has acquired. The various distributions of the framework can also enable data analytics to take place over different levels. Data acquired by one facility may be used in an other as long as the contextual requirements are met. This enables data analytics to be performed on a much larger scale.

The next chapter will present an implementation of this framework.

Chapter 4

Gamification Framework Implementation

In this chapter the implementation of the gamification framework will be presented. The details of the implementation of the six main components: the crowdsourcing dashboard, the sensor interface, the gamification, the event detection, the event labelling and the data analytics will be described.

As mentioned in Chapter 3, there are two main use cases for the implementation of the framework: one oriented towards acquiring a dataset for research and the other to acquire labels to obtain insights from sensor data. The implementation presented in this thesis focuses on the latter. Additionally, there exist two main types of sensors that are targeted by the framework; portable sensors and permanent sensors. Portable sensors are those that can be moved and that measure various types of data and objects. They are not permanently installed in a specific location nor are they bound to measure specific objects. Permanent sensors on the other end are installed in a fixed location and are intended to measure the same objects and quantity on a continuous basis. Although the framework and its implementation supports both types of sensors, this particular instance is oriented towards permanent sensors. It is important to note that each of the components of the framework was implemented and that the implementation does support any scenario the framework was designed for. However, in order to provide a more focused direction for our work, the examples and the case study presented will focus on the idea of deploying the framework with the intention of acquiring labels for existing infrastructure sensor data.

The concept behind this implementation of the framework is to enable administrators of facilities where sensors are installed to setup a competitive environment to reach a common goal directly related to the sensors. An example of such competitive environment could be to create teams and communities to battle against one another to see who is able to reduce their energy consumption the most. The framework would be deployed for a specified period of time in order to motivate the users to change their behaviour while allowing for the capture of enough sensor data labels to perform analytics. The idea is that anyone taking part in the game shall join communities. Each community has specific requirements either in terms of the users location or to their access to specific types of sensors. Each of those communities will contain a specific set of actions directly related to community requirements and its common goal. If a user meets the requirements, they may join the community and participate in the game. Basically, communities will represent the labelling environment of the framework. They define what type of data is to be labelled and in what context, as well as the labels to be apposed.

For the sake of portability and maintainability, the framework was implemented using a service oriented architecture. Figure 4.1 depicts the implementation in terms of services.

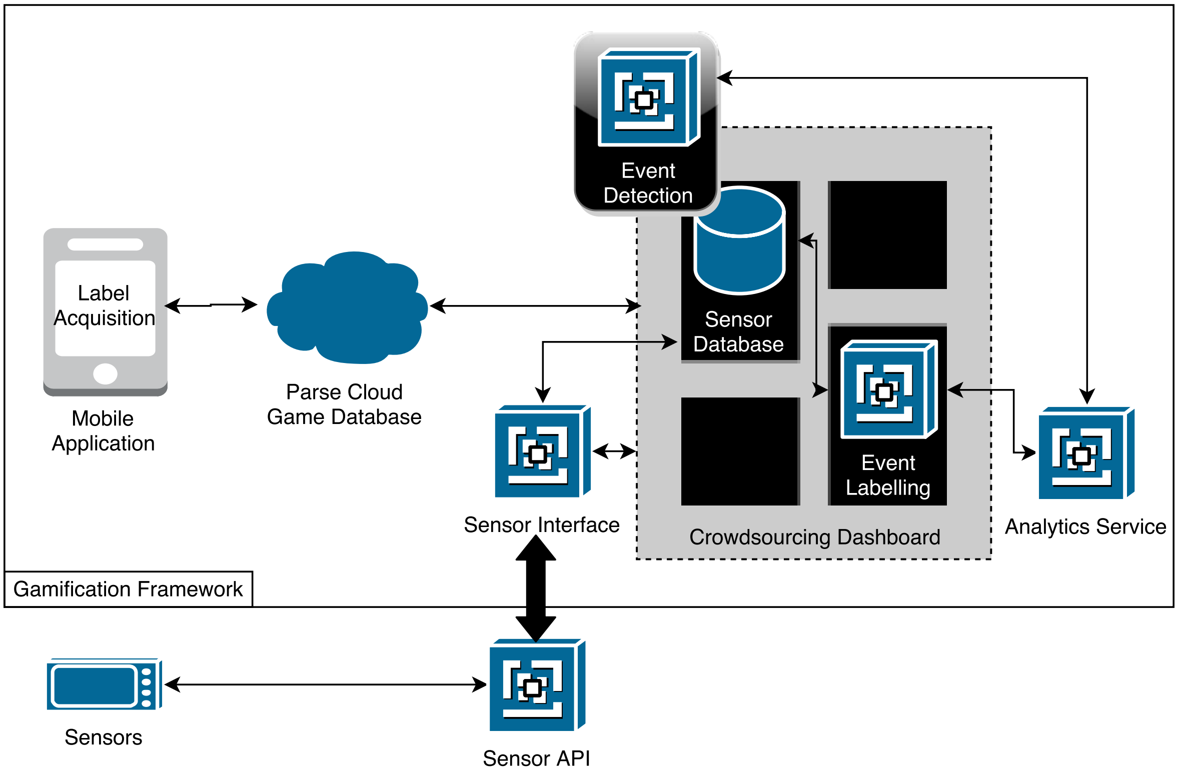


Figure 4.1: Implementation of the Gamification Framework.

In order to facilitate the deployment of the framework across many possible gaming application, the game database was implemented using Parse [74], a cloud based database which is setup to communicate through an API. Conversely, the event detection module was implemented using database functions and triggers whereas the event labelling module was implemented as a service within the crowdsourcing dashboard. This specific implementation will be discussed in this chapter.

4.1 Crowdsourcing Dashboard

The crowdsourcing dashboard is designed as the central piece of the framework. It is used to receive and send data to the other components of the framework. The *game_backbone* entity is implemented as the community artifact. Each community will have specific requirements in terms of sensors, locations, and measurement types. The community shall also own a set of specific sensors and it shall describe where each of the sensors are installed. Therefore each community may have a different set of gaming and sensor requirements. The crowdsourcing dashboard shall enable administrators to setup those communities in order to participate in the game. The specifics of the implementation of the dashboard will be discussed next.

4.1.1 REST API

In order to facilitate the communication amongst components, the dashboard was implemented as a REST API in Java [75] and Scala [76] using the Play framework [77]. The dashboard was implemented as a secured web application that enables administrators to login, create a community and setup all the required parameters through the use of forms. This implementation focuses on permanent sensors, therefore the physical context was implemented using a locational hierarchy shown in Figure 4.2.

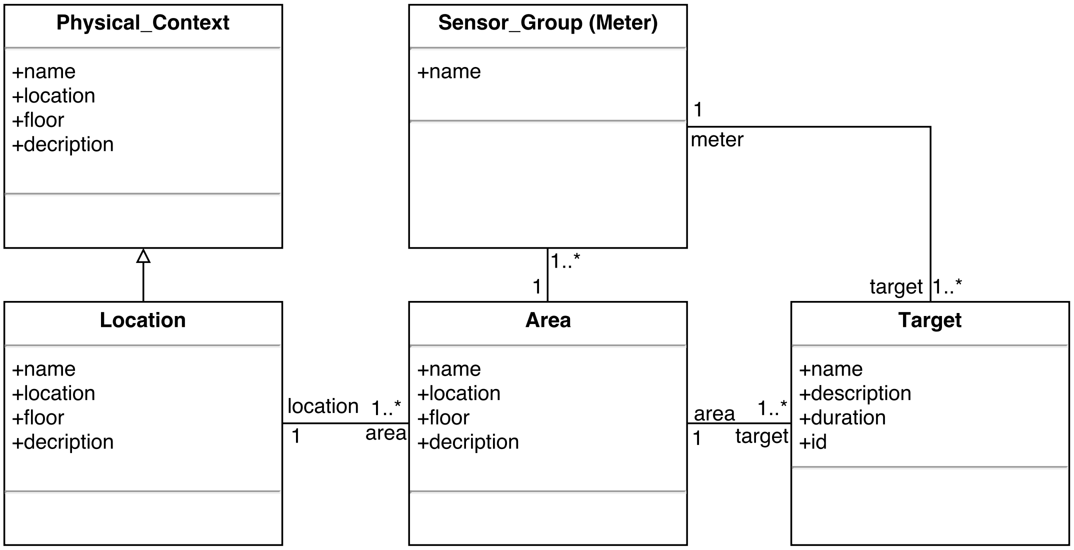


Figure 4.2: Implementation of the Physical Context.

Administrators are responsible for creating or associating communities with their specific sensors in order to enable users to join them and participate in the game. In this implementation, we are dealing with meters. The meters represent a group of sensors measuring the same objects. It is analogous to the sensor group entity of our design.

During the setup, the administrator will have the option of creating a community. Once that community is created, the administrator must create the location object in which they will provide a description of the location where the meters are installed, this will include the address, the number of floors etc. Typically, locations are divided into areas within a building. This is often done to sub-meter a building, for example a different set of sensors may be measuring the first floor and the second floor. In order to properly translate sensor readings into labels, we need to know how the sensors and meters are related to their physical environment. The administrator will be asked to create areas within their location through the use of a form. The form as shown in Figure 4.3 will ask for a description of the area, the floor or wing it is in, etc.

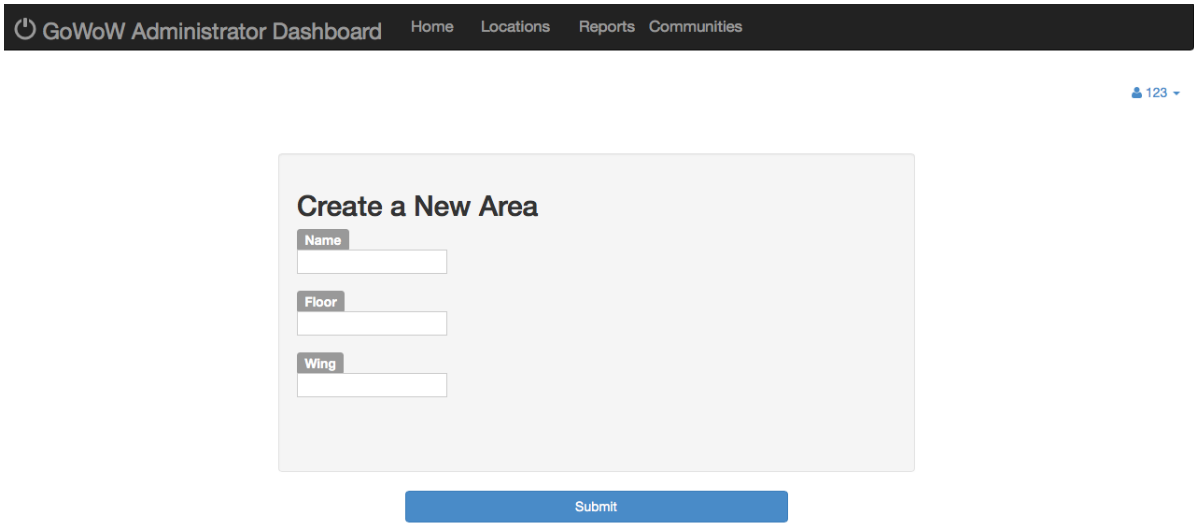
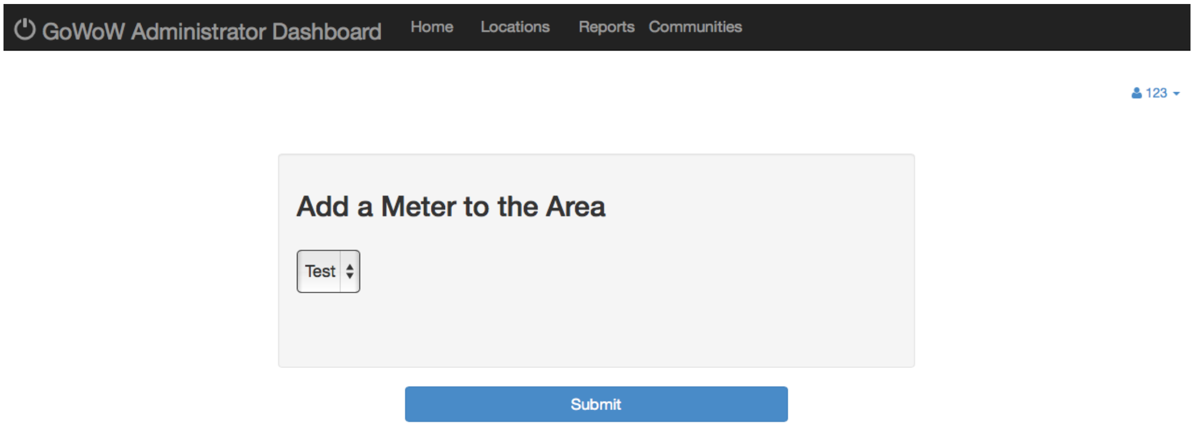


Figure 4.3: Area Creation Form.

Once the areas are created, the administrator shall add the meters that are installed within each area. The administrator will create the meter from a list of pre-established sensors and associate them with those areas as shown in Figure 4.4.



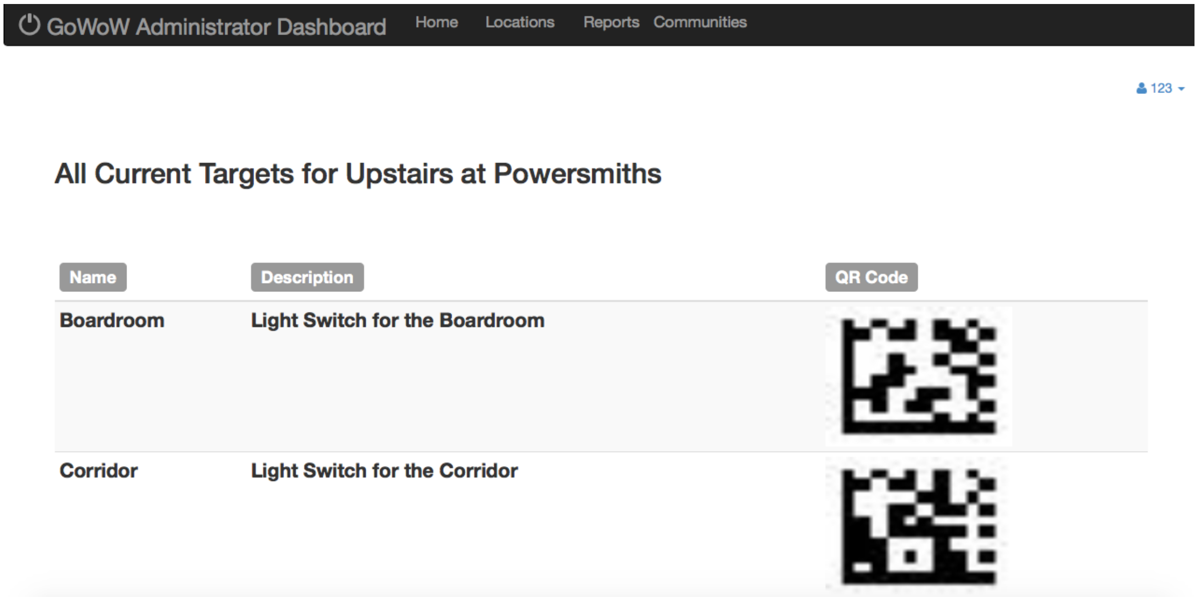
Add a Meter to the Area

Test

Figure 4.4: Meter and Area Association Form.

Lastly, the administrators shall also create actions they wish the user to accomplish. Once actions are created, they must also create targets which are a very specific location, object or duration related to the action to be accomplished, therefore a more precise label. For example, an action may be to turn off the light and the target will be the actual light switch. Each action may have multiple targets. Those targets should be directly associated with the appropriate sensor. This will enable the translation of physical actions into sensor reading labels.

When creating a target, the administrator will select from a dropdown menu the area (and therefore the sensor) the target is associated with. Upon the creation of a target, a QR code is automatically generated. This QR code is the chosen means to enable action capture. By creating the QR code on the crowdsourcing dashboard the administrator will be able to print them and install them to facilitate gameplay. Figure 4.5 shows the list of targets and their QR code within our case study.



All Current Targets for Upstairs at Powersmiths

Name	Description	QR Code
Boardroom	Light Switch for the Boardroom	
Corridor	Light Switch for the Corridor	

Figure 4.5: Target QR Code View.

The web application enables the capture of the data required to create the contextual entities such as the locations, areas, sensors, meters, actions and targets. Once created, the entities are saved to the relevant databases: the sensor database or the game database. The following section will outline the sensor database schema.

4.1.2 Sensor Database Design

The sensor database is implemented to support the design of the framework. Figure 4.6 shows the ER diagram of the sensor database tables related to the crowdsourcing dashboard. It can be seen that the community table, is representative of the *game_backbone* entity while the relationships between the sensors and the gaming targets are facilitated through the meter area table amongst others.

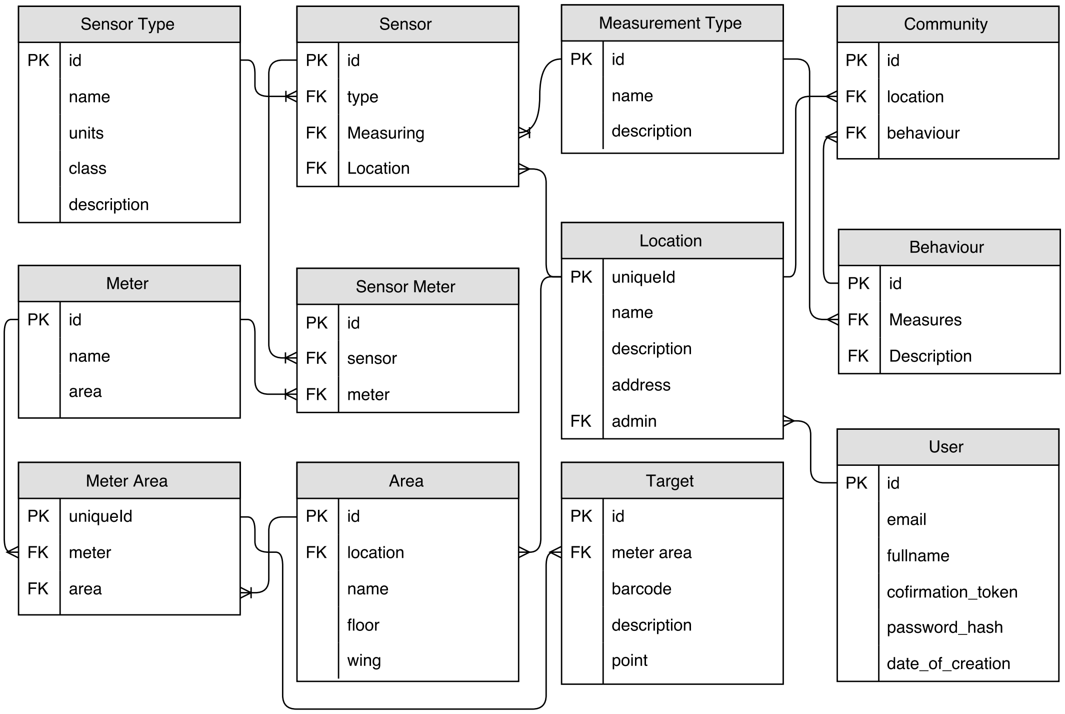


Figure 4.6: Sensor Database ER Diagram for Dashboard Entity.

In the sensor database, the sensor_group object is represented by the meter entity. Meters are indeed a physical representation of a group of sensors measuring the same targets. The sensor database implementation will be further discussed in section 4.4.

4.2 Sensor Interface

As a means to interface with the sensors, a flexible interface following a factory pattern was designed using a service oriented architecture. This will allow for the translation from sensors readings to persistent objects.

In order for the sensor data to conform to the format of the reading object upon which the framework relies, we need to create an interface that will enable us to communicate with a multitude of heterogenous datatypes. Different adapters were designed to enable the exchange and conversion of the data to the required format. The design of the sensor interface is presented in Figure 4.7.

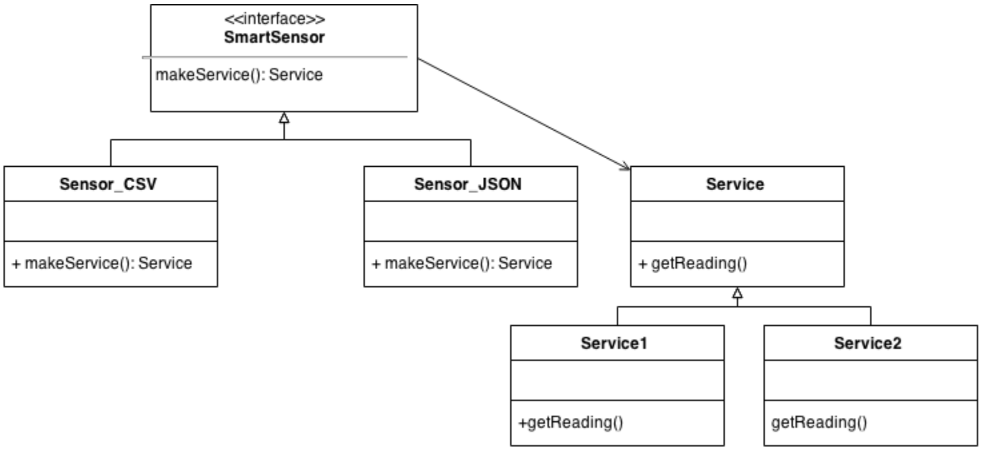


Figure 4.7: Sensor Service Factory.

Within the crowdsourcing dashboard, the administrator will have the ability to select the type of data provided by the sensors, the factory design will enable the use of the proper class implementation. Furthermore, the administrator will identify the tags or keys used by the sensors which will be mapped to the proper format through a form selection. A new type of service class will then be generated using the API address provided within the dashboard form. This design enables the administrator to easily deploy the solution to any type of sensor data. Once the information has been gathered, the administrator may deploy the solution. The deployment process is shown in Figure 4.8. The Figure depicts how the proper SmartSensor interface implementation is instantiated and the service generated.

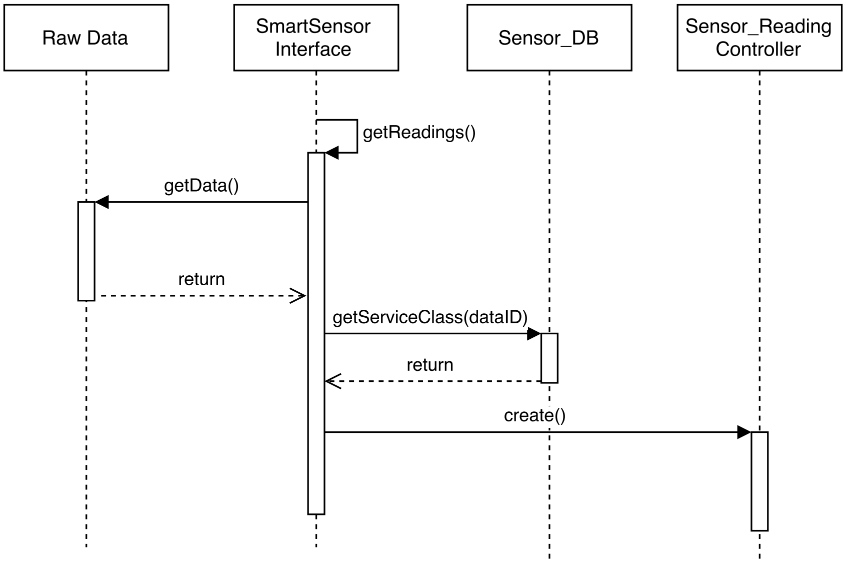


Figure 4.8: Sensor Service Creation Process.

Once the service has been generated, the service will query the data on its own. The data gathering process is shown in Figure 4.9. It is worth mentioning that the interval at which the data is queried is left for the administrator to decide. The framework is designed to ideally handle a real time sampling frequency.

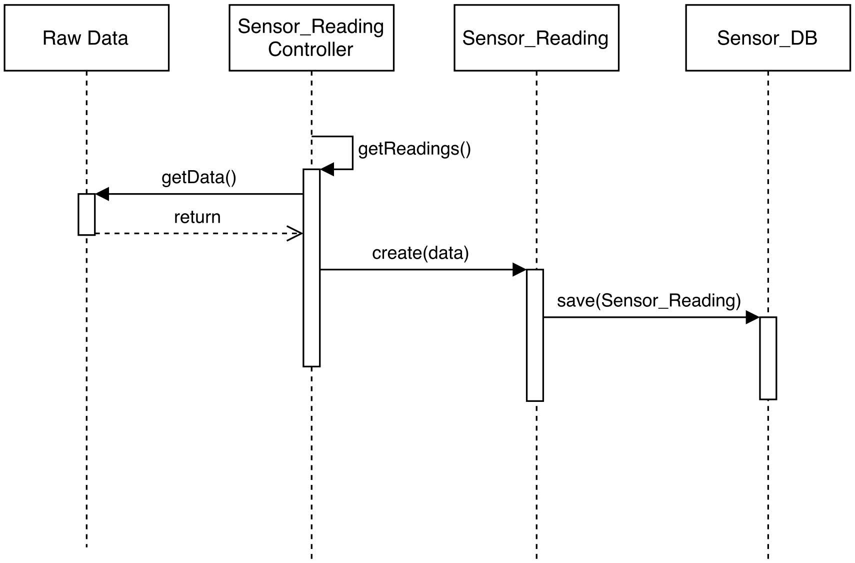


Figure 4.9: Sensor Data Gathering Process.

4.3 Gamification

The idea behind this implementation is to create a game based on competitions amongst communities. It sets out an environment where users can join communities and participate by accomplishing actions and missions. The goal is to encourage behavioural changes at the individual and community level. The application could be used by large organizations who wish for their users to change their habits. Users will compete against each other and against other teams within the organization to demonstrate their commitment and success in changing their habits.

Gamification relies on four main factors: Mechanics, Measurement, Behaviour and Reward [64] . The game application was designed to integrate all of these components in order to leverage the full power of gamification. Figure 4.10 is adapted from Bess' [64] design and presents each of the gaming elements along with the specific instances used in this implementation.

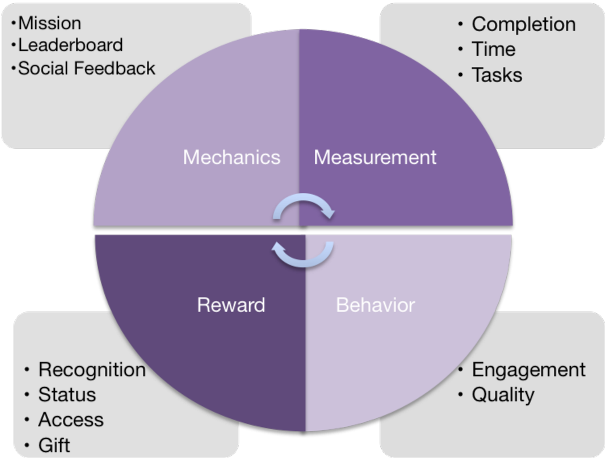


Figure 4.10: Gamification Elements.

As per the framework design, the application implements four different types of artifacts. We will describe the implementation chosen for each of the mechanics, rewards, behaviour and measurement artifacts.

Various gaming mechanics are incorporated within the game. First of all, the application was implemented using a level hierarchy. Each level corresponds to a certain amount of points. In order to motivate the users to continue, each level contains a different set of actions. The more the user progresses and levels up, the more actions are unlocked. As the actions are created in the dashboard, the administrator shall select which level they wish to make this action available to, as well as the points reward for the accomplishment of this action. Secondly, a leaderboard can be viewed to motivate users to surpass others. The users can observe their standing within their community and team. The measure of success in this case is measured by the number of tasks performed by the users.

The reward artifact will be created when the targets are generated in the crowdsourcing dashboard. Administrators shall select the multiplier value for the accomplishment of an action at this target. Leaving the control of the target multiplier to the administrators enables more control over which target may need to be favoured for analytical or goal accomplishment purposes. Additionally, administrators may grant physical rewards or gifts to the users that are best performing according to the results from the leaderboards.

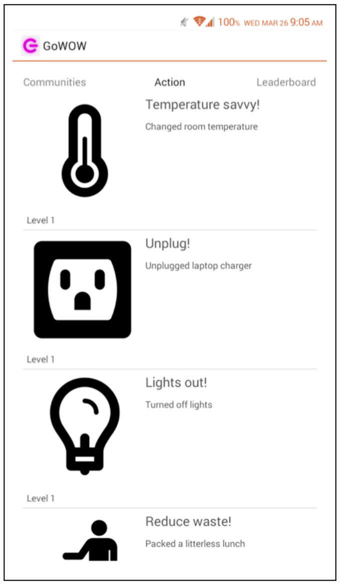
The behaviour artifact can be customized by the type of action created by the administrator. It is represented by the action theme and game behaviour tables in the database. It is inferred by the action categories.

The measurement artifact is observed through the completion of tasks by the users. The rate at which users are performing tasks and the ratio of completed tasks will serve as the measure of success of the gamification.

4.3.1 Game Application

The game application, a mobile game on the Android platform, was partly implemented as a part of a fourth year design project in 2014 by Caglioti, Detriech, Emonds and Smith [78]. The original application was modified in order to support the design of this framework. Changes were made particularly to the supported actions, the action capture methodology, the reward system, the action records and the game database.

The mobile application implements various gaming mechanics. The level hierarchy was built within the application with a set number of levels and point targets. The calculation of user points and rankings was also directly implemented within the application. However, the action and target selection are kept completely separated from the implementation and are rather pulled from the database thereby allowing for easy changes of the game content. Figure 4.11 shows an example of a list of actions available to the user along with a view of the scrollable menus at the top of the application.



100% WED MAR 26 9:05 AM

GoWOW

Communities Action Leaderboard

Temperature savvy!

Changed room temperature

Level 1

Unplug!

Unplugged laptop charger

Level 1

Lights out!

Turned off lights

Level 1

Reduce waste!

Packed a litterless lunch

Figure 4.11: View of the Game Application.

The application allows the users to perform many functions such as: join communities, view leaderboards and perform actions. Details of the implementation in regards to the label acquisition and game database will be discussed in the following subsections.

4.3.2 Label Acquisition

The main purpose of the mobile application is to capture actions to enable labelling. Therefore, the mobile application must allow for the recording of gaming events. In order to ease the use of the application, for this implementation QR codes were chosen as the optimal way of selecting the action/target pair. A very important factor of gamification is to ensure a great user experience. If the user is required to make many menu selections to play the game, the usability and fun factor would be greatly affected. In order to maintain the playability of the application, a QR code scanner was integrated within the design to remove from the user the hassle of multiple menu selections. This QR code will be used by the mobile application as a means to validate that the action was completed.

Additionally, by placing QR code next to the desired target we can ensure a more valid response from the users. The framework relies highly on the good faith of its users. However, due to the challenge and reward nature of the game some safety mechanisms are required to ensure that the users are truly participating. Having to physically scan a QR code is likely to ensure that the user actually did perform the task they claimed as opposed to having to simply click on a menu item. Figure 4.12 depicts the process involved in recording an action.

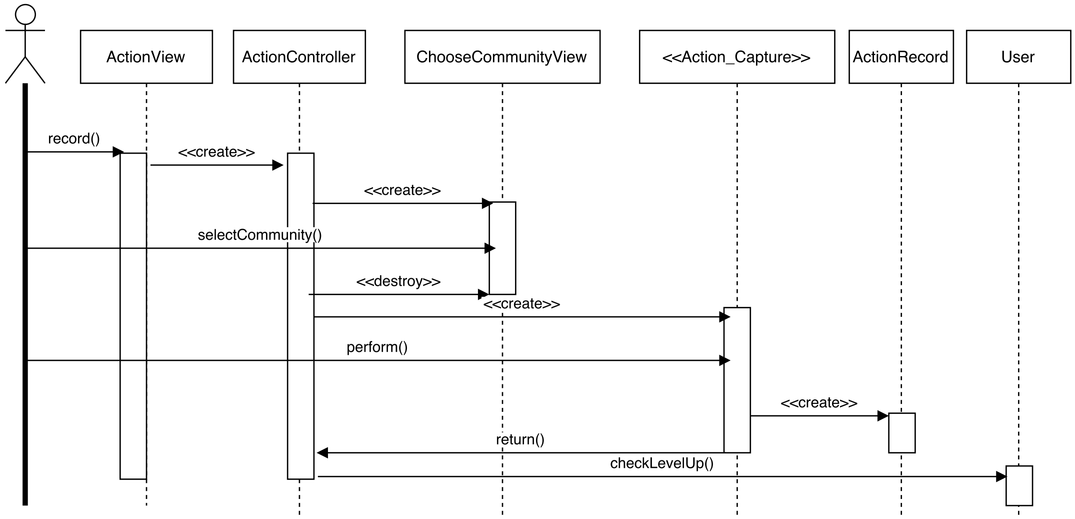


Figure 4.12: Action Record Sequence Diagram.

In order to motivate the users to level up, actions available to higher levels are shown but are not accessible. This serves as an incentive for the user to keep progressing. Once actions are accomplished, they are recorded and sent to the game database.

4.3.3 Game Database Deployment

In order to facilitate the access to the game from any location or platform, a cloud database was chosen to deploy the game database. The Parse database was chosen for various reasons. First of all, an Android library was available to quickly and effectively exchange data securely to and from the database. Additionally, a REST API was available to enable any other module to access the data. Lastly, the security layer was built into the services provided by the database which highly facilitated the development of the mobile application.

The game database is an essential part of the gamification component, it was designed as the central access point for all of the components of the gamification architecture. It serves as the middle ground between the crowdsourcing dashboard and the game application. By keeping the gaming data separated from the actual application we are enabling quick development for other platforms as well as easy updatability and maintainability.

The database is used for two main purposes, one to hold all the game actions and target data as set out in the crowdsourcing dashboard. The mobile application will simply pull the required action set for a users community and display it based on the users' level. Figure 4.13 depicts the process:

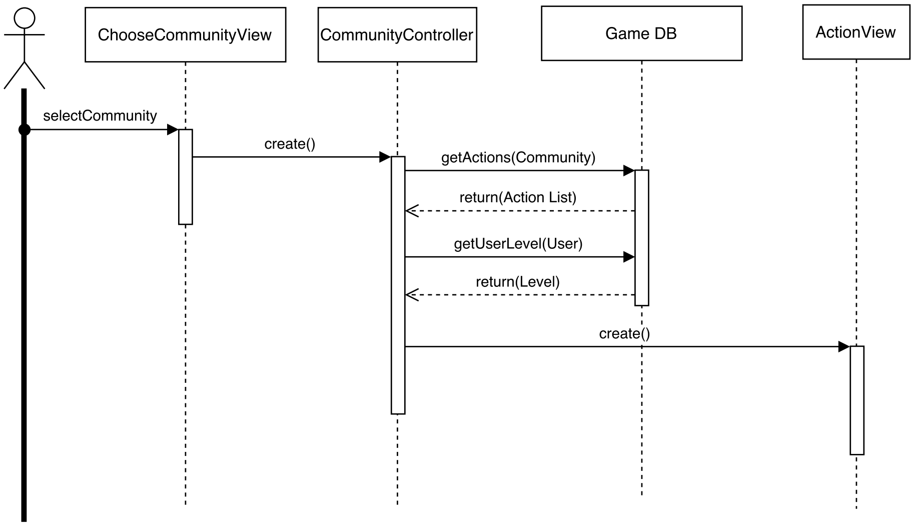


Figure 4.13: Action Sourcing Sequence Diagram.

The second purpose of the database is to save the gaming events; once actions are performed as described in Figure 4.13 an Action Record is created and saved within the database. This will enable the crowdsourcing dashboard to merge readings and labels in order to perform analytics.

The ER Diagram implemented in the Parse database is shown in Table 4.14.

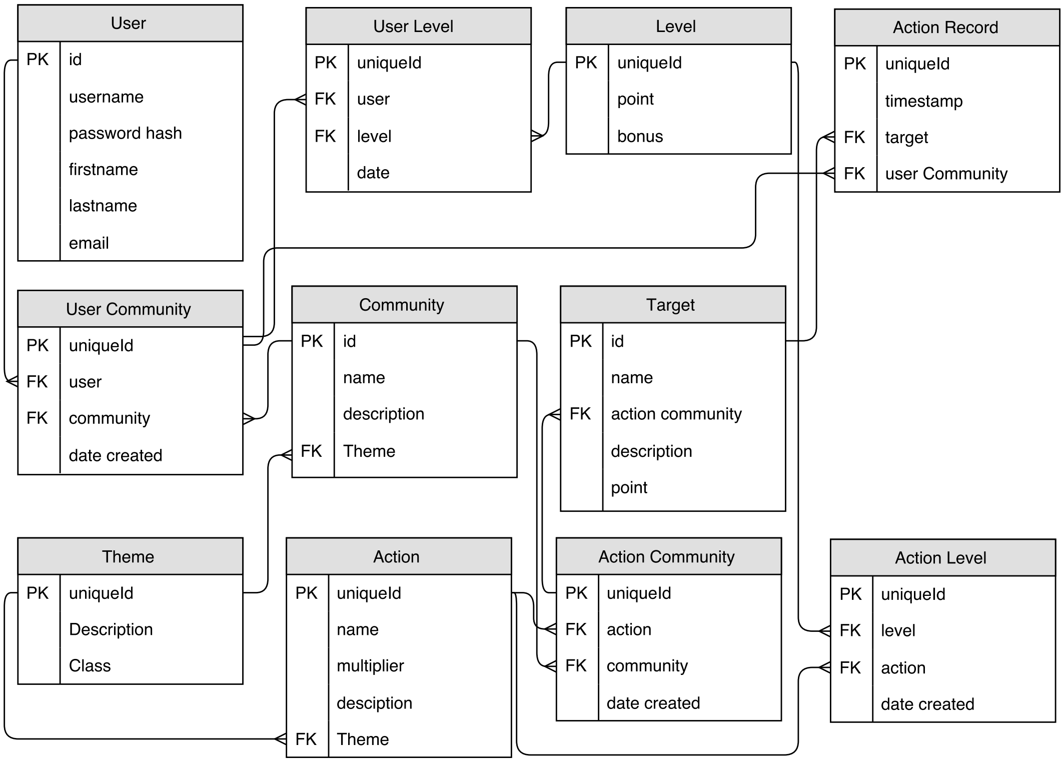


Figure 4.14: ER Diagram of the Game Database.

4.4 Event Detection

In order to support real time event recognition, it was decided that the event detection module shall be implemented as database triggers. Figure 4.15 represents the ER diagram of the event detection and event labelling components of the framework; it may be used for reference purposes within the next subsections.

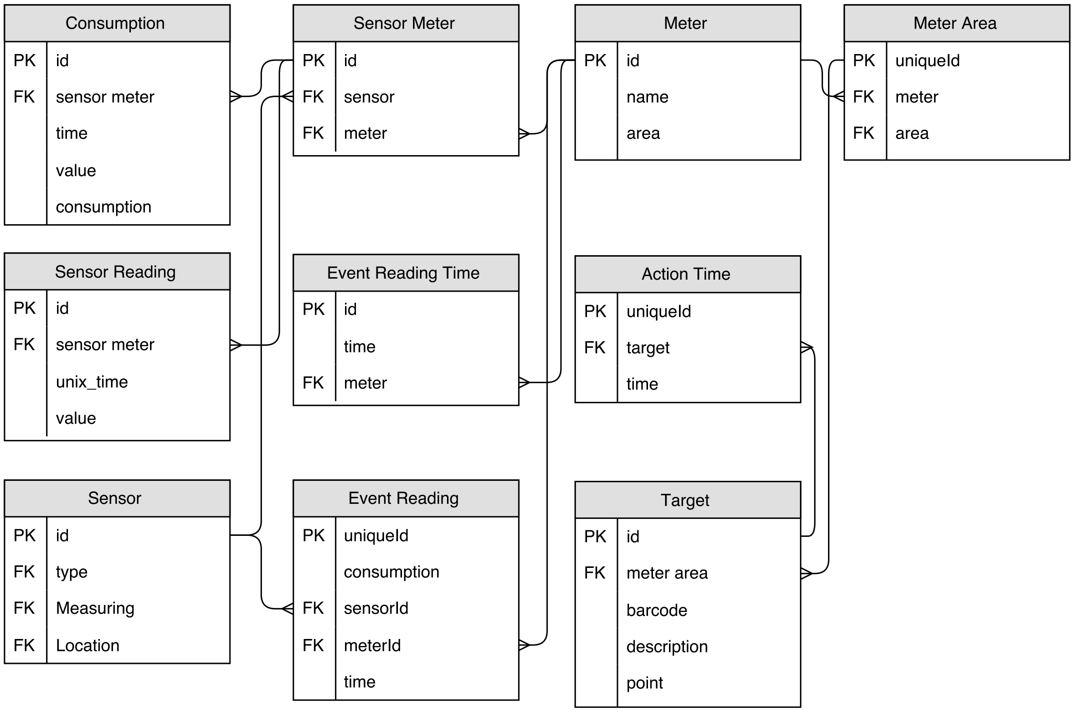


Figure 4.15: Sensor Database ER Diagram for Event Detection and Labelling.

The event detection module can be separated into two process: the pre-processing and the event detection. Both of these processes were implemented as database triggers and will be discussed in the following subsections.

4.4.1 Pre-Processing process

In order to implement the pre-processing process, a database trigger was designed. The notion is that as the sensor interface inserts a reading from a specific sensor, the trigger would automatically calculate the difference between the current reading and the previous reading from the same sensor. This new value would then be inserted into the consumption table. This would represent the change between readings that we are interested in. Listing 4.1 shows the after insertion trigger where the sensor ids we are monitoring are between max_sensorID and min_sensorID.

Listing 4.1: Pre-Processing Change Detection Trigger

```
CREATE DEFINER=‘root‘@‘localhost‘
TRIGGER ‘meterdata‘.‘data_reading_AFTER_INSERT‘
AFTER INSERT ON ‘data_reading‘ FOR EACH ROW
IF(NEW.idnum<max_sensorID AND NEW.idnum>min_sensorID) THEN
INSERT INTO gowow.consumption (readingId,time,value,consumption)
```

```
VALUES (NEW.idnum, NEW.unix_time, NEW.value,((
SELECT
    NEW.value-g1.value
FROM
    data_reading g1

WHERE
  g1.idnum=new.idnum AND g1.unix_time<new.unix_time AND
  g1.id<>new.id
    GROUP BY unix_time
  ORDER BY unix_time desc
    limit 1)));
END IF
```

As the pre-processing step gets completed, the event detection takes place. This process will be depicted in the following subsection.

4.4.2 Event detection process

For each insertion within the consumption table, we are monitoring to see if the reading is greater than the rolling mean for that sensor. If that is the case, we insert the timestamp into the eventReadingTime table. However, this is only done if an event was not already detected for another related sensor. If an event was already recognized for this timestamp the process is halted. Listing 4.2 provides the code for the event detection trigger as an after insertion trigger.

Listing 4.2: Event Detection Trigger

```
CREATE DEFINER=‘root‘@‘localhost‘
TRIGGER ‘change_AINS‘ AFTER INSERT ON ‘consumption‘
FOR EACH ROW
BEGIN
IF(New.readingId<max_sensorID and New.readingId>min_sensorID)
THEN
  IF(abs(new.consumption)>getRollingMean(New.readingId)
    THEN
      IF((select count(*) FROM eventReadingTime WHERE
      time=new.time)<1)
        THEN
          INSERT INTO eventReadingTime(time) VALUES (new.time);
          INSERT INTO eventReading(time) VALUES (new.time);
        END IF;
    END IF;
END IF;
```

```
END
```

It was determined experimentally that the rollingMean tends to plateau due to the extremely large number of very small changes. Therefore, for experimental purposes the getRollingMean(New.readingId) was replaced with a fixed value after a period of 1 week of readings. This greatly improved the real time performance of the algorithm.

Once a reading is inserted within the eventReadingTime, all of the readings from the same sensor group shall be combined and inserted within the eventReading table. Listing 4.3 provides the code for the event insertion trigger as an after insertion trigger of the eventReadingTime table.

Listing 4.3: Event Insertion Trigger

```
CREATE DEFINER=‘root‘@‘localhost‘ TRIGGER
‘meterdata‘.‘eventreadingtime_AFTER_INSERT‘ AFTER INSERT ON
 ‘eventreadingtime‘ FOR EACH ROW
 BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE ids INT;
  DECLARE temp DOUBLE;
  DECLARE cur CURSOR FOR SELECT m.sensorId FROM meterSensor m
  WHERE m.id = NEW.meterId;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
  OPEN cur;
        ins_loop: LOOP
            FETCH cur INTO ids;
            IF done THEN
                LEAVE ins_loop;
            END IF;
            SELECT consumption INTO temp from consumption
            WHERE time=new.time
            AND readingId=ids;
            INSERT INTO eventReading
            VALUES (null,NEW.time,NEW.meterId,ids temp,null);
        END LOOP;
 CLOSE cur;

END
```

This particular code was written using the labels set out by the administrator for our case study which will be presented in the upcoming chapter.

4.5 Event Labelling

The event labelling module had to be implemented using a combination of manual and automated processes. The gaming event and the sensor events were paired based on the timestamp plus a time sliding window. Therefore, if a gaming event was labelled at 10:03:33 and the sensor sampling frequency was 30 seconds, then the event data would be matched with the label if an event was detected within a 30 second window. Experimentally, it was established that there is up to 2 seconds between the time the action is physically accomplished and the time the QR code was scanned. Therefore the sliding window will take place 2 second prior to the record and 28 seconds after. The following query was used to appropriately merge the streams and label the event data.

Listing 4.4: Event Labelling

```
SELECT *  FROM (SELECT
    from_unixtime(a.time/1000) as time_a,
  from_unixtime(b.time/1000) as time_b,
    c.time as time_c, e.time as time_e,
  a.target as tar_a,b.target as tar_b,
  c.value, c.sensorId
  FROM
   action_time  AS a
LEFT JOIN
   action_time  AS b
      ON f
      rom_unixtime(a.time/1000)<from_unixtime(b.time/1000)
      AND
      from_unixtime(a.time/1000)>
      (from_unixtime(b.time/1000)-interval 29 second)
LEFT JOIN
   consumption c
   ON (from_unixtime(a.time/1000) - interval 2 second < c.time
   AND
  c.time<from_unixtime(a.time/1000)+interval 29 second)
JOIN
  eventreadingtime e
    ON c.time=e.time
ORDER BY from_unixtime(a.time/1000)) as te
WHERE te.time_a not in (
  SELECT from_unixtime(g.time/1000) as tim
     FROM action_time f LEFT JOIN action_time g
     ON from_unixtime(f.time/1000)<from_unixtime(g.time/1000)
          AND f
rom_unixtime(f.time/1000)>
(from_unixtime(g.time/1000)-interval 29 second )
WHERE from_unixtime(g.time/1000) IS NOT NULL
)
    AND te.time_c IS NOT NULL
```

Once the labels have been apposed, a clustering algorithm was manually used on each of the labelled sets. The *k*-means algorithm was used in an attempt to separate the data into clusters, one representing appropriately labelled data and the other representing labels in which we are unsure. This process was performed in case other actions not recorded by the game occurred during the sampling time, this enabled us to discard those actions as invalid.

4.6 Analytics Module

The analytics module was implemented in two ways. First of all, the analytics module provided real time evaluation of events. More specifically, as events were detected by the framework, the analytics module performed a classification in order to provide some insight about what was happening within the facility in real time. The classification was based upon the data labels previously acquired. Secondly, historical analytics were provided using the acquired data labels to extract knowledge from data that had been previously gathered. The analytics module was implemented using R [79] and R Shiny[80].

4.6.1 Real Time Analytics

R Shiny was responsible for monitoring the event table within the database. Whenever a new event was inserted, the data was captured in R and a classification was performed using *k*-nearest neighbour, where the learning set of the algorithm was the game labelled data. The result of the classification, that is the labels that were assigned to each readings, were sent back to the sensor database for further analytical purposes. There are multiple reasons behind the use of *k*-nearest neighbour but it was mostly due to its versatility and ease of use. Indeed, as targets are added or modified, new training labels can easily be added and modified without requiring extensive retraining. Due to the changing nature of the framework, this was critical.

4.6.2 Historical Analytics

The second analytical implementation was used to extract information from past data using newly acquired labels. Reports were created and generated using R Shiny based on the users selection of the day they wished to learn information from. The historical data for those days was then inserted in the data reading table if it was not already present. The event detection was then generated and classification was performed on those events to provide historical insights on the facility being monitored.

The implementation was then used for a case study, the results of which will be presented in the following chapter.

4.7 Summary

In summary, this chapter presented an implementation capable of robustly capturing sensor data labelling using gamification. The implementation presented followed the gamification framework architecture presented in Chapter 3. It made use of an implementation of the *game_backbone* entity to acquire all required components to connect the sensor environment to the gaming environment. In this case, the community entity encompassed the metadata.

The mobile application implemented in Android pulled all the required gaming information from the community entity. It fetched the required actions, rewards, levels, etc. Only the view of the actual gaming mechanics was implemented directly in the mobile application. The crowdsourcing dashboard, developed using the play framework, enabled the setup of the sensor and gaming parameters by asking the user to create locations, meters, areas, actions, targets, rewards etc. therefore populating the gaming metadata. Additionally, the four gamification building blocks were implemented as follow:

- The gaming mechanics chosen for this implementation were the leaderboard, points and task accomplishment mechanics.
- The rewards were implemented in terms of points and bonuses but also through social feedback with the help of the leaderboards.
- The actions and communities were designed to be built around a behavioural change or theme.
- The measurement of the success was implemented through the user levels and labelling conversion rate.

By implementing the chosen gaming mechanics, rewards, behaviours and measurements we are showing how the framework could have been implemented in a variety of games.

The event detection and event labelling modules were implemented as database functions and triggers in order to detect the event as effectively as possible and support the real-time requirement of the framework. By using database triggers, the events were detected upon the insertion of the sensor readings in the sensor database.

The data labelling clustering as well as the various analytical components were developed using R and only provide a glimpse of what is possible to accomplish with the real-time detection of events and label acquisition.

The work presented in this chapter only serves as a proof of this concept and is solely an example of what can be accomplished using the framework.

Additionally, the chosen implementation is such that researchers could also setup their sensor environment and leverage the power of crowds to obtain a labelled dataset. Although it was not explicitly used in this fashion, the manner in which the use case could be fulfilled is evident.

Chapter 5

Gamification Framework Evaluation

In this chapter the evaluation of the gamification framework will be presented. The gamification framework will be evaluated in the context of a case study for Powersmiths, a company specializing in manufacturing electrical sensor meters, transformers and software solutions for industrial buildings. The chapter will describe how the framework was used to acquire data labels and perform real time and historical data analysis. The methodology used to evaluate the event detection, labelling accuracy and the analytical capabilities of the framework will also be detailed. The results of the analytics will also be discussed

5.1 Case Study

A case study was performed by the author to establish the value and accuracy of the gamification framework. Powersmiths [81] is a company developing meters capable of measuring various electrical consumption features at very fast intervals. They have been capturing and storing this data for a number of years but to this point had only used the data in order to get a real time snapshot of consumption. The company had an underlying interest for sustainability and energy savings and wished to see if some analytics could be performed using their data. This presented a great opportunity to evaluate the proposed framework.

The company was more specifically interested in finding out about their lighting habits. For example, they wanted to know if they would be able to establish which lights were turned on and where, as well as, whether or not they had sustainable habits within their facility in regards to lighthing. The ability to extract such knowledge would add much more value to the data that they were already storing.

The crowdsourcing framework was used to setup their gaming environment. One location, their head office, was created along with a single area: the second floor of the facility. The head office, has a number of meters installed, but a single meter is responsible of measuring the consumption of the second floor lighting. Therefore, this meter was created and added to the area. Powersmiths was interested in monitoring all light switches on the second floor for a total of 11 different targets. Those targets were created and the QR code printed and installed next to the light switches.

For this particular implementation, a single measuring action, turning off the lights, was created. However, a number of other sustainable actions were also added; such as reducing waste and unplugging various devices as described in Table 5.3. Those actions though, were not captured or monitored by the framework but were essential to the success of the game. Indeed, a variety of actions is necessary in order to keep the users entertained.

In addition to the electrical sensors, Powersmiths also had a number of occupancy sensors installed in the same areas. Those sensors were fabricated by SmartThings [82], a home automation technology manufacturer. Powersmiths was interested in joining the sensor data to perform further analytics and establish whether or not energy was being wasted due to lights being left on while rooms were not occupied. The SmartThings occupancy sensors were then added to the framework and monitored simultaneously. However, no specific gaming actions were created for their labelling purpose as the sensor data was already labelled. Figure 5.1 shows the overall architecture of this specific implementation.

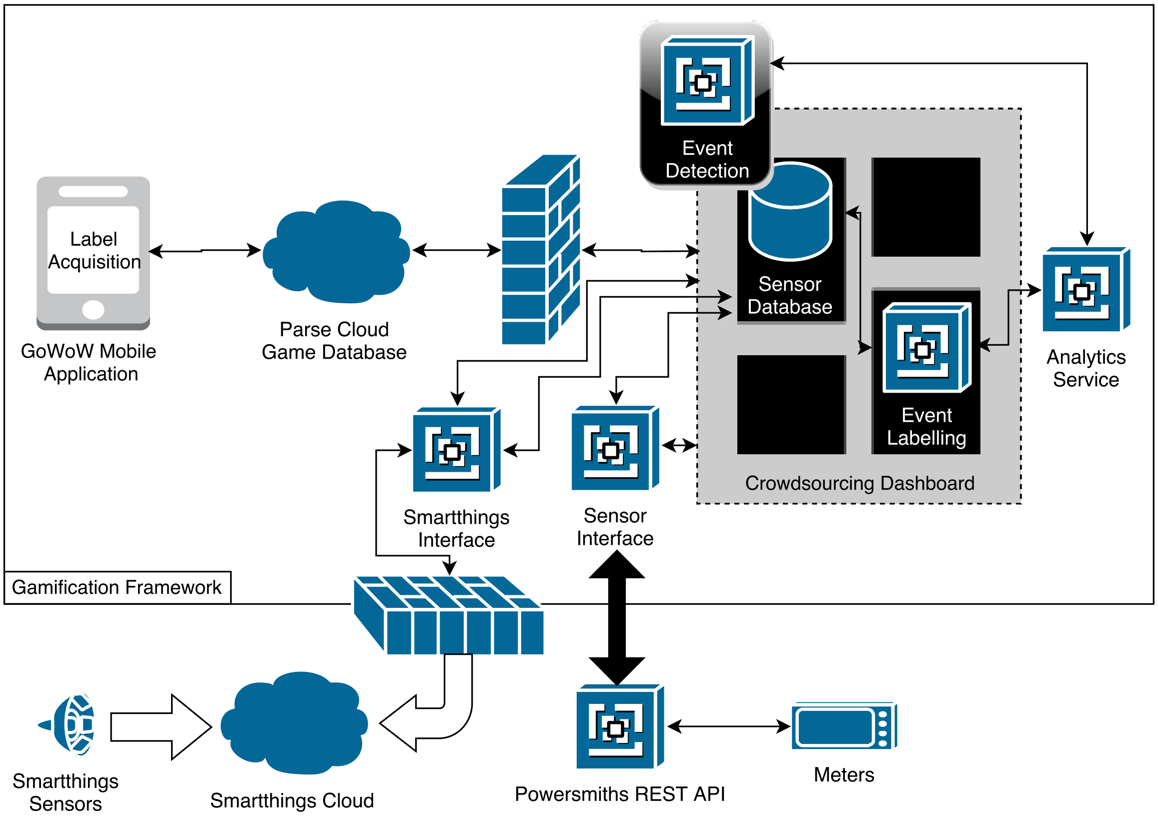


Figure 5.1: Implementation Design.

The electrical meters are composed of various sensors each measuring different electrical features of the lighting system. The various electrical sensors are presented in Table 5.1

	SensorName	*Units*	*Description*
1	Amp A	Ampere (A)	Measures the amp of the phase 1 of current
2	Amp B	Ampere (A)	Measures the amp of the phase 2 of current
3	Amp C	Ampere (A)	Measures the amp of the phase 3 of current
4	Amp N	Ampere (A)	Measures the amp of the neutral phase of current
5	Amp L	Ampere (A)	Measures the amp of the effective current
6	kW A	Kilowatt (kW)	Measures the kilowatt consumption on phase 1
7	kW B	Kilowatt (kW)	Measures the kilowatt consumption on phase 2
8	kW C	Kilowatt (kW)	Measures the kilowatt consumption on phase 3
9	kW total	Kilowatt (kW)	Measures the total kilowatt consumption

Table 5.1: Electrical Sensors Descriptions.

Powersmiths sensors are designed to constantly measure the values but the readings are only updated and made available through an API every 30 seconds. However, Powersmiths experienced some technical difficulties with their API during the 82 days of the case study. Given the sampling rate, we expected to record 2880 readings daily from each of the sensors. In reality, on average each sensor recorded 2470 sensor readings within a 24 hours period. During our experiment, only the meter responsible for the second floor lighting was of interest. This meter contained 9 sensors as shown in Table 5.1, for a total of 22,230 readings on average per day for the meter. The framework was implemented and data gathered from the meter for 82 days, accumulating over 202,587 readings on average per sensor for a grand total of 1,823,289 readings.

On the other hand, the SmartThings sensors were also queried every 30 seconds and the motion values were recorded only when the status of a sensor had changed. The motion sensors are described in table 5.2

$SensorName$	$Units$	$Description$
VP's office	Motion True or False	Measures whether motion occurred in the VP's and project manager's Area, enabling us to infer occupancy.
Boardroom	Motion True or False	Measures whether motion occurred in the Boardroom, enabling us to infer occupancy.
Support	Motion True or False	Measures whether motion occurred in the Support Area, enabling us to infer occupancy.
Forge	Motion True or False	Measures whether motion occurred in the Forge enabling us to infer occupancy.
Corridor	Motion True or False	Measures whether motion occurred in the Corridor Area, enabling us to infer occupancy.

Table 5.2: Occupancy Sensors Descriptions.

The game application was deployed with the following set of actions described in Table 5.3. A number of actions were not sensor based, however these actions are essential for the success of gamification. Users enjoy diversity and challenges and in order to achieve a change in behaviour a variety of actions is required.

ActionName	*ActionCategory*	*Level*	*Description*
Lights Out!	Lighting	1-2	Turn any of the targeted lights off!
Reduce waste!	Environment	1	Packed a litterless lunch
Temperature savvy!	HVAC	1	Changed room temperature
Unplug!	Outlet	1	Unplugged laptop charger
Reduce waste!	Environment	2	Refilled a water bottle
Temperature savvy!	HVAC	2	Turned off air conditioning, replaced with fan

Table 5.3: Game Application Actions.

Only the Lighting System was of interest for this experiment, therefore only the light switches were targeted. The targets can be found in Table 5.4.

TargetName	*TargetId*	*Description*
VP	1	This is represents the light switch for the VP's office
Project Manager	2	This is represents the light switch for the Project Manager's office
Boardroom	3	This is represents the light switch for the Boardroom
Corridor	4	This is represents the light switch for the Corridor Area
Forge Total	5	This is represents the light switch for the Entire Forge Room
Forge 1	6	This is represents the light switch for the front of the Forge Room
Forge 2	7	This is represents the light switch for the back of the Forge Room
Support	8	This is represents the light switch for the Support Area
VP and Project Manager	9	This is represents the light switch for the VP's and Project Manager's offices
Support and Corridor	10	This is represents the light switch for the Support and Corridor Areas
Project Manager and Boardroom	11	This is represents the light switch for the Boardroom and the Project Manager's office

Table 5.4: Lighting Action Targets.

It can be observed that some targets are a combination of two targets, the reason behind this approach is that some lighting switches are found side by side. Therefore, there exists a strong possibility that one would turn off/on both switches at the same time. These targets aim at capturing those behaviours explicitly.

The evaluation of this framework was twofolds, one part consisted in evaluating the ability of the framework to detect events and appropriately label them and the second evaluated its analytical capabilities.

5.2 Event Detection and Labelling Component Evaluation

The gaming experiment was conducted over a single day, enabling the capture of 134 action records using the mobile application with the help of 2 users. Unfortunately, the trial was limited to a single day due to hardware limitations within the facility. In actuality, the office is highly oriented towards apple products and since the application was developed using Android devices, loaner phones had to be used.

During the experimentation, it was noticed that the sensor API was experiencing technical difficulties, it was not serving data consistently or regularly and the 30 seconds sampling interval was not kept. Indeed, of the expected 2880 daily readings from each sensors only 1808 complete readings were recorded, a sampling rate of 62.8%. During the game play between 10 am and 2 pm the accuracy was slightly higher at 65.1%. As a result, of the 124 actions, the framework accurately captured only 90 corresponding readings, for a sampling accuracy percentage of 72.5%. Given the technical difficulties that led to variable and infrequent data sampling, this accuracy is considered successful. The reason for the inability to capture some of the events was that data was simply not available from the API at the time the actions were completed. The initial dataset size is therefore 90 readings.

5.2.1 Event Detection Evaluation

In order to test whether the framework would enable the identification of false positives through the labelling of readings recorded when actions were captured but not physically performed, 10 targets were scanned without the actions being physically completed. Of those 10 actions, the framework detected 9 associated readings. Which means that one of the action although recorded did not have any related sensor readings and was therefore discarded. Of the remaining 9, all 9 were not detected as events and therefore the framework was able to fully prevent false positives. Table 5.5 shows the injected readings and demonstrates the validity of our event detection approach. Going forward, we wish to remove the 9 newly injected false readings from the dataset, reducing the initial dataset from 90 to 81 readings.

$ActionTime$	$ReadingTime$	$EventTime$	kWA	kWB	kWC	$kWtotal$
’2014-12-16 10:02:36’	’2014-12-16 10:03:02’	NULL	’0.000’	’0.000’	’0.000’	’0.001’
’2014-12-16 10:52:27’	’2014-12-16 10:52:32’	NULL	’0.000’	’0.001’	’0.001’	’0.002’
’2014-12-16 10:58:21’	’2014-12-16 10:58:32’	NULL	’0.000’	’0.001’	’0.000’	’0.001’
’2014-12-16 11:15:33’	’2014-12-16 11:16:02’	NULL	’-0.000’	’-0.000’	’-0.000’	’-0.001’
’2014-12-16 11:45:57’	’2014-12-16 11:46:02’	NULL	’-0.000’	’-0.000’	’-0.000’	’-0.000’
’2014-12-16 11:52:51’	’2014-12-16 11:53:02’	NULL	’0.000’	’-0.000’	’-0.001’	’-0.001’
’2014-12-16 12:10:04’	’2014-12-16 12:10:32’	NULL	’-0.000’	’-0.001’	’-0.001’	’-0.002’
’2014-12-16 12:13:04’	’2014-12-16 12:13:32’	NULL	’0.000’	’-0.001’	’-0.001’	’-0.002’
’2014-12-16 13:20:33’	’2014-12-16 13:21:02’	NULL	’-0.000’	’-0.001’	’-0.001’	’-0.001’

Table 5.5: False Positive Detection.

For the purpose of the experiment, we are interested in capturing a single gaming action per sampling interval. However due to the inconsistent sampling rate, of the remaining 81 actions captured and recorded it was found that 4 occurred during the same sampling intervals. Those samples were discarded leaving us with 77 unique samples to label.

5.2.2 Event Labelling Evaluation

Additionally, each of the sensor target had to be clustered to judge whether other actions may have occurred during the sampling or whether the users may have mislabelled the actions. Therefore, if it is judged to be significantly different, it should be excluded from the labelled set. This evaluation was performed using R, a statistical analysis program, through *k*-means clustering. The accuracy of the clustering was measured using the BSS/TSS ratio. All the readings for a specific target were clustered using one or more centres. The number of centres was increased until the BSS/TSS ratio was over 80%, indicating a proper dispersion. The readings that did not appear to belong to the same cluster as the majority of the readings were then excluded. Figure 5.2 shows how two of the readings were removed from the labelled pool for target 3.

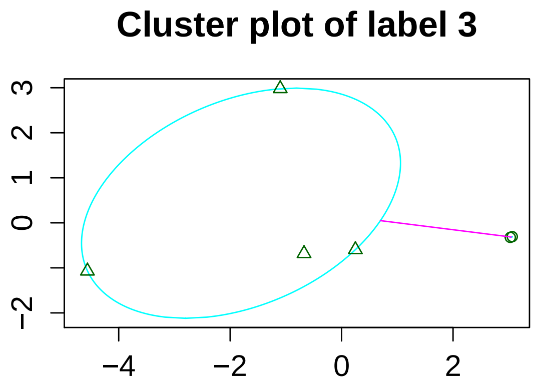


Figure 5.2: Cluster of Target 3.

In order to evaluate the framework's ability to prevent noise introduction, 5 invalid samples were injected by performing a labelled action but scanning the improper target. The goal being to evaluate whether the framework would accurately identify them as invalid labels. The framework was successful in recognizing 4 out of the 5 injected actions. However, the other action had no readings associated with it and therefore was not captured through the API. Table 5.6 shows those results. It can be observed that the clustering algorithm placed the noisy readings in a different cluster than the cluster containing the properly labelled readings, leading to the rejection of the injected readings.

ActionTime	*ReadingTime*	*TargetNumber*	*AssignedCluster*	*TrueCluster*
2014-12-16 12:55:29.692000	2014-12-16 12:55:32	12	3	1
2014-12-16 12:58:25.051000				
2014-12-16 13:02:18.226000	2014-12-16 13:02:32	10	2	3
2014-12-16 13:05:26.451000	2014-12-16 13:05:32	10	1	3
2014-12-16 13:33:55.048000	2014-12-16 13:34:02	4	1	2

Table 5.6: Wrong Label Detection.

Therefore, after removing the 4 recognized injected actions, we are left with a set of 73 valid detected actions. Which means that if every single one of the detected action that we did not tamper with were to be labelled, the labelled dataset should contain 73 readings.

However, this is not the case. A number of factors may have introduced noise within the dataset, especially given the improper sampling time. After the clustering is performed for each of the targets, a number of potentially noisy data readings were removed. The composition of the final set is shown in Table 5.7

Target	*Number of Labels*
'1'	'5'
'2'	'6'
'3'	'4'
'4'	'4'
'5'	'4
'6'	'6'
'7'	'6'
'8'	'5'
'9'	'7'
'10'	'3'
'11'	'3'
Total	53

Table 5.7: Final Dataset Composition.

Therefore the framework translated 72.6% of the gaming actions into labels. However, we expect that the percentage would be much higher if the API was to be functioning adequately.

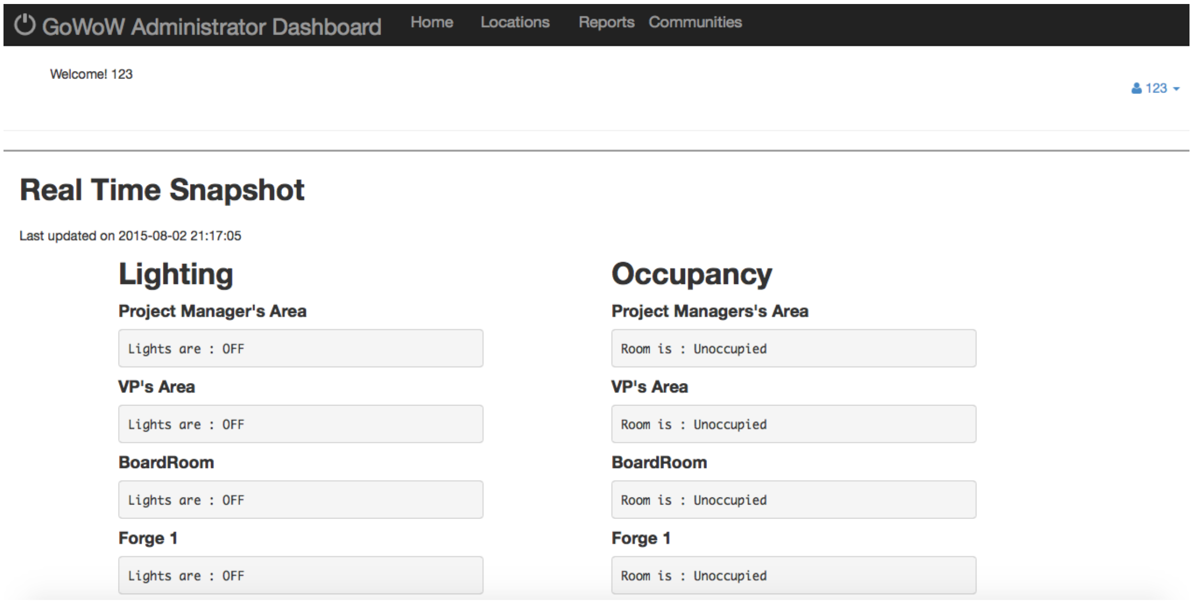
The following section will depict how the framework leverages those labels to perform sensor data analytics.

5.3 Analytical Evaluation

The analytical capability of the framework enabled through the data label acquisition was evaluated both in real time and for historical data.

5.3.1 Real Time Analytics

The real time analysis was evaluated through the use of a R Shiny application, deployed through the gamification framework. The application was designed to monitor the lighting and occupancy of the Powersmiths' facility and as events were taking place, the status of each area was shown. The application is presented in Figure 5.3.



Real Time Snapshot

Last updated on 2015-08-02 21:17:05

Lighting

Project Manager's Area

Lights are : OFF

VP's Area

Lights are : OFF

BoardRoom

Lights are : OFF

Forge 1

Lights are : OFF

Occupancy

Project Managers's Area

Room is : Unoccupied

VP's Area

Room is : Unoccupied

BoardRoom

Room is : Unoccupied

Forge 1

Room is : Unoccupied

Figure 5.3: Real Time Analysis Application.

The idea behind the real time analysis was based upon *k*-nearest neighbours. The acquired data labels were used to create clusters and from there the centroid of each target was calculated. In order to keep the complexity of the algorithm low to support the real time results, the newly detected event data was compared only against the centroids and assigned to the class it was closest to. Therefore, limiting the computation to a maximum of 12 comparisons, where 11 is the number of potential labels plus the non-event label.

Although informal, this aspect of the framework was evaluated by ensuring that the information made available by the application was accurate. This was done over multiple days, for a period of approximately 30 minutes per day. Due to the fact that the framework was deployed at an out of town location, the facility administrator was contacted periodically and the accuracy of the results was verified. Indeed, lights were purposely turned on and off and the results were recorded and compared against the real time snapshot provided by the application. Once again the API providing information from the sensor was not consistently accurate. However, during each interaction, each of the lights were turned on/off once and the experiment was repeated. The accuracy results were found to be 80%.

In order to evaluate the accuracy and value of the framework, multiple analytical comparisons were performed against state of the art algorithms, each used to label the identified events.

The accuracy metric used is defined by Equation 5.1:

$$Accuracy\_kmeans = (tp + tn)/total \tag{5.1}$$

Where:

tp represents the number of true positives in the dataset

tn represents the number of true negatives in the dataset

$total$ represents the total number of samples in the dataset

It is possible that unsupervised classification is sufficient for the scope of this work and that perhaps the labelling does not provide any significant advantages. In order to demonstrate the validity of our approach comparisons using the k-nearest neighbours algorithm in R and the k-means algorithm within the Weka [83] knowledge analysis tool were performed. Three comparisons were made:

- A first attempt was made to classify the raw data readings identified as events by our framework using the k-means algorithm in Weka. In an unsupervised unlabelled context, the only information that would be accessible to the user would be the number of potential clusters or targets. Therefore this information was provided to the algorithm. The same computation was performed using both the raw event readings and the pre-processed event readings. This comparison shows the importance of the pre-processing step.

- Secondly, a comparison of the same approach was performed but this time the clustering noise removing technique was applied to both data sets. This comparison shows the importance of the noise removal.

- Thirdly, R was used to demonstrate the importance of having access to labelled data. A comparison between the previous unsupervised comparisons and the use of a very simple k-nearest neighbour algorithm is made. We simply chose the training set to be equal to the centroid of each of the cluster labels and set the number of neighbours to one. This approach is as simple as classification can be and enables us to most closely compare against the previously unsupervised k-means approach.

Table 5.8 shows the results of the various classifications of the event readings. The accuracy enhancements related to the use of labelling and data processing are non-negligible.

Case	*Algorithm*	*Accuracy*
Unprocessed Event Readings No Noise Removal	K-Means	31.08%
Pre-Processed Event Readings No Noise Removal	K-Means	64.86%
Unprocessed Event Readings	K-Means	35.84%
Pre-Processed Event Readings	K-Means	71.69%
Pre-Processed Event Readings and Labeled Centres	K-Means	88.67%

Table 5.8: Framework Comparison with Unlabelled Techniques.

Lastly, using a combination of the algorithms evaluated herein and the occupancy sensor data, historical analysis was performed. The following section will evaluate this work.

5.3.2 Historical Analytics

Through the combination of the data obtained by the occupancy sensor and the ability to classify detected events with high accuracy as shown in the previous section, historical analytics were performed in order to extract insights from electrical usage data.

Daily historical reports were created to provide a snapshot of the usage of the facility. Within each daily report, every single event detected is listed and classification is performed to determine which event occurred. The detection and classification of the events, enables us to extract the lighting status of each area. Since we can also determine the occupancy status of the area through the use of the motion sensors, it can be established whether the facility is wasting energy at any given time as described in Table 5.9

Light	*Occupancy*	*Waste*
ON	ON	NO
ON	OFF	YES
OFF	ON	NO
OFF	OFF	NO

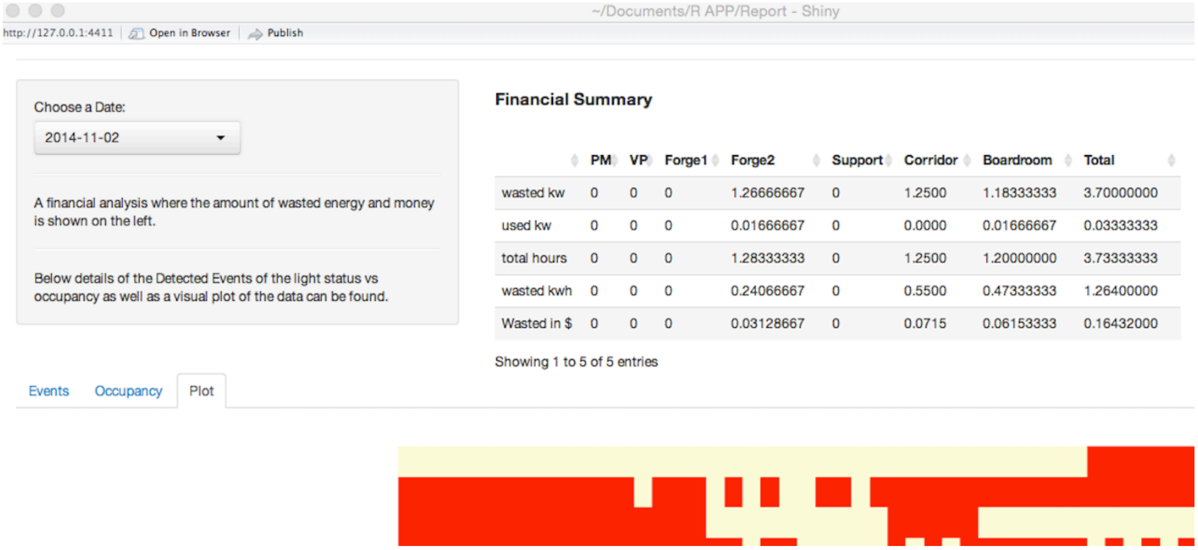
Table 5.9: Electrical Consumption Waste Rules.

These rules enabled us to create a financial report which detailed the amount of energy, the duration and the cost of the waste. The application was developed using R Shiny and was integrated within the crowdsourcing dashboard. A view of the application is shown in Figure

5.4. The entirety of the data captured by the framework, 1,823,289 readings, were analyzed by this component.

The evaluation of this component was once again performed through informal interviews with the facility administrator. The administrator confirmed the accuracy of the reports and commented on how the reports enabled him to effectively change some wasteful consumption habits. An area was identified as particularly wasteful and the daily reports truly helped to remediate the situation. Therefore, these reports enabled the administrator to track the effectiveness of gamification towards behaviour changes. It was remarked that the game had a positive effect and that the waste observed within the VP's area was almost completely eliminated since the implementation of the framework.

Figure 5.4 depicts the view of the report.



Choose a Date:

2014-11-02

A financial analysis where the amount of wasted energy and money is shown on the left.

Below details of the Detected Events of the light status vs occupancy as well as a visual plot of the data can be found.

Financial Summary

	PM	VP	Forge1	Forge2	Support	Corridor	Boardroom	Total
wasted kw	0	0	0	1.26666667	0	1.2500	1.18333333	3.70000000
used kw	0	0	0	0.01666667	0	0.0000	0.01666667	0.03333333
total hours	0	0	0	1.28333333	0	1.2500	1.20000000	3.73333333
wasted kwh	0	0	0	0.24066667	0	0.5500	0.47333333	1.26400000
Wasted in $	0	0	0	0.03128667	0	0.0715	0.06153333	0.16432000

Showing 1 to 5 of 5 entries

Figure 5.4: Historical Analysis Application.

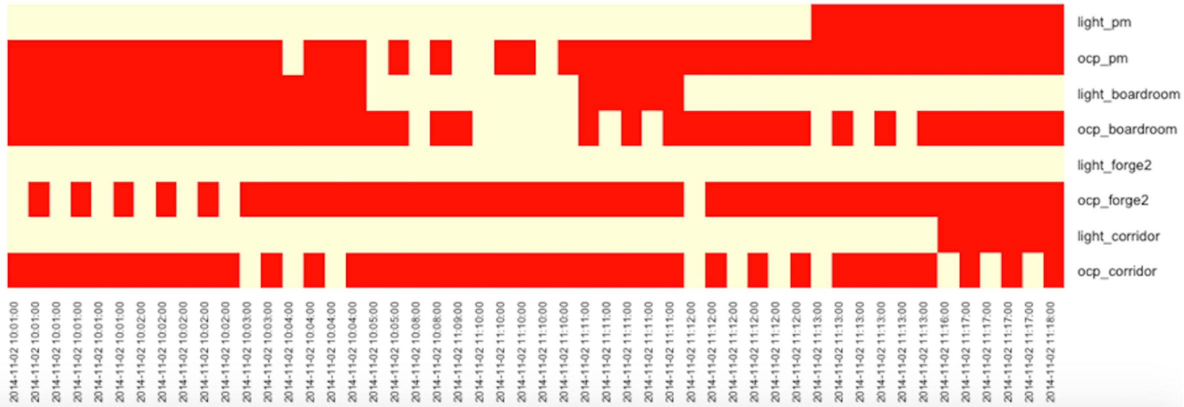


Figure 5.5: Historical Analysis Application Heat Map View.

The plot found in Figure 5.5 is simply rendition of the lower half of Figure 5.4. It is a heat map representing the waste for each areas. Indeed, as found on the right end side, for each paired line a contrasting colour indicates waste in accordance with Table 5.9. If the line appears pale yellow, this signifies that the room is occupied or that the light is on depending on the line you are looking at. Conversely, red indicates an absence of occupants and that the lights are off. Therefore, if the line representing the light of an area is yellow but occupancy is red, energy is being wasted.

5.4 Summary

In summary, over the course of one day and with the participation of only two users, a dataset of 53 hard labels was successfully gathered through the use of the gamification framework. It is to be noted that the game was played during the workday and therefore the users were still performing other duties during that time. Given the technical limitations faced by the framework, we consider this labelling rate to be a success.

The accuracy of the labelling technique and the analytical capabilities of the framework were also evaluated. Through the injection of invalid data, the labelling technique was tested in terms of its resistance to false positives and its ability to accurately identify true data events. By simulating the improper labelling action we were able to evaluate the frameworks autonomous ability to prevent the introduction of noise in its labelled dataset. The framework was hugely successful in detecting all invalid actions taken by the users. We consider the evaluation of the robustness of the labeling technique to be a great success.

In terms of real time and historical data analytics, the results were evaluated through the use of human feedback. The administrator of the case study reported great results in terms of real time analytics with an average of 80% accurate event detection.

Additionally, the historical trends detected by the framework signalled that some areas of the building were consistently wasting energy. With the use of the framework, the users were able to positively change their behaviour and the changes were accurately reflected by the frameworks reporting tool. The real time and historical analytical capabilities of the framework are also considered a great accomplishment.

Chapter 6

Conclusion and Future Work

This chapter will provide a review and conclusion of the contributions, implementation and evaluation of the gamification framework for sensor data analytics presented in this thesis. Lastly, a description of the intended direction and future work related to this framework will be presented.

6.1 Conclusion

The work presented in this thesis provides a solution to the problem of sensor data labelling and presents various contributions:

- The framework makes use of gamification as a crowdsourcing paradigm to cope with both the high cost associated with the acquisition of sensor data labels and the burden faced by the user that is typically associated with performing sensor data labelling tasks. In fact, users are participating in a game rather than explicitly labelling data, henceforth encouraging true participation and limiting user introduced annotation errors due to boredom or lack of interest. The manner in which gamification is integrated and central to the framework is considered a contribution of this thesis. The gamification component is integrated within the design of the framework as opposed to other approaches that use gamification as a secondary tool. Typically gamification is implemented as a fixed game that is completely separated from the solution's architecture. It is often used as nothing more than an interesting user interface. This work proposes a solution to deploy any game to label any type of sensor data.

- The novel data labelling approach presented here utilizes a multi channel architecture to collect sensor data and labels separately. This separation is what enables the framework to support any type of sensors, whether mobile or fixed. The framework was designed to be adaptable to any sensor data source which in itself serves as a contribution.

- The framework allows the real time detection and analysis of the sensor data. By effectively separating the capture of sensor events from the gaming human activities, the framework is able to provide real time and historical data analytics. The capture of sensor data is completely independent from the labelling process which means that whether users are currently playing the game or not, sensor data is being captured and analyzed. The solution leverages the labels previously gathered to provide real time and historical sensor data event classification. Therefore providing an end to end solution to the issues of data analytics related especially to sensor data gathered within permanent infrastructure.

- The shortcomings of current crowdsourcing frameworks used to label datasets are addressed by this thesis, in response to those weaknesses associated with sensor data. The work here proposes an approach to label sensor data not only to provide data analytics but also to enable researchers to easily obtain labelled datasets.

Furthermore, in order to provide a proof of concept for the gamification framework the implementation of the various components was presented. This thesis includes the implementation details of:

- The crowdsourcing framework as a REST API along with the reasoning behind its implementation strategy as a Web Service.

- The sensor interface and its flexible architecture that enables the support of various data types.

- The gamification component including how the mobile application and database schema implemented the gamification strategy presented by the framework.

- The data pre-processing step and its underlying logic. A discussion on the idea of extracting the anomalous properties of the sensor data to showcase events was also detailed.

- The event detection module which included how consideration was given to respect the real-time requirements of the framework.

- The data labelling module which made use of data polling services, database queries and clustering techniques to ensure robust data labelling.

- The analytical services including how the services were made reactive to real time data and how historical analysis was performed using the labels obtained through gamification.

Additionally, the implementation of the sensor labelling framework was evaluated on multiple facets. The evaluation was performed through the use of a case study. A game was developed and utilized with the purpose of labelling data captured by permanent sensors and performing both real-time and historical analysis.. The subsequent components were evaluated as follow:

- The ability of the framework to detect events was evaluated by having users play the game and evaluate how many of the gaming events were actually captured by the framework. Given the technical difficulties faced by sensors which affected the consistency of the sampling rate, the framework was capable of detecting the majority of the events it was presented with.

- The capacity of the framework to reduce noise was also evaluated by the injection of invalid data labels. Gaming errors were purposely introduced to test the framework's ability to remove user introduced data noise. The framework was successfully able to catch the events that were purposely mislabelled along with gaming events that were submitted without the users physically accomplishing the task. Therefore the framework successfully demonstrated its ability to reduce noisy data.

- The aptitude of the framework to provide both real time and historical data analysis was also evaluated through repeated testing and visual validation. Indeed, remote testing was performed with the help of a facility administrator to validate the real time results shown within the application. A retrospective analysis was also performed regarding the historical analysis results. The analytical capability were positively confirmed through this process.

With the increasing presence of the IoT in our everyday life and the enormous amounts of sensors found in everyday devices, an easily adaptable means to acquire intelligence from those devices is required. The gamification framework addresses the need to translate data into actionable information by providing a solution that can easily be implemented in any context such as: smart building, health care or even wearable devices. The integration of gamification within our flexible multi channel architecture opens the door to the combination of crowdsourcing and crowdsensing techniques. However, consideration shall also be given to the security and privacy issues that could arise from the deployment of such framework. The utilization of our framework in the context of finance or military applications could also enable to extraction of potentially privacy damaging information.

With the emergence of the Big Data revolution, a surprising number of people are interested in extracting insights from their own personal data. A number of platforms provide those services to the user, for example smart electrical meter data can be uploaded to a variety of services to gain insight on consumption habits and a variety of wearable devices provide users with information in regards to their sleeping and physical activity patterns. However, these services are commercial services which often require the purchase of specific devices and the analytical insights they provide are often superficial and limited.

The work in this thesis provides the stepping stone to a universal way to gather and label any form of data and render data analytics accessible to the masses.

6.2 Future Work

The modular architecture of the framework renders the addition of future work easily feasible. Each of the components of the framework can easily be replaced with various other modules as more research is performed and better solutions become available. The following ideas could be explored as future work for the framework:

- Due to the limitation of the available data during our case study, our work could highly benefit from more robust testing. Subjecting our framework to a longer case study would allow us to get a better measure of the accuracy and value the framework. Additionally, by extending the case study the effectiveness of the long term deployment of gamification could be observed. The basis of our trust model could then be evaluated based on the participation of more users and conclusions could be drawn regarding the truthfulness of the participants. Furthermore, by extending the case study to different sensor interfaces, the impact of imperfect data could be evaluated.

- In order to reduce the number of labels required to perform data analysis, active learning algorithms are often used to pick the best data labels upon which algorithms should be trained. This type of algorithm typically queries the user to obtain specific labels rather than asking the user to label the entire dataset. It can be described as targeted labelling. Active learning is very difficult to achieve when it comes to sensor data because the events of interest cannot be translated from sensor data to human activity without having prior labels. The idea of using readings similarity to identify a labelled reading that is most similar to the reading of interest and then to request from the user to perform the associated activity may be explored. This is an interesting area as it could vastly improve the quality of the dataset gathered by our framework. Gamification would not only be used to provide sensor labels but also to request them, therefore it would fully capitalize the user engagement enabled by the very integration of gamification.

- The architecture of this framework was built to handle real time data. However, no consideration was given to the amount of data nor to the potential high velocity it may reach. Nonetheless, the event detection techniques made use of a computationally inexpensive methodology to facilitate the deployment of the framework in a Big Data environment. Adapting the current architecture to handle Big Data constitutes the first priority in the direction of this work due to the close connection between sensor data and Big Data. The use of the lambda architecture [84] to support real time and historical data analysis as well as data labelling in the context of Big Data has been explored and the implementation of this paradigm has begun.

- Integration of our work with the concept of semantic web is an other area of interest. Using semantic web within our framework would enable the integration of heterogenous sources of data which could allow us to create more complete datasets in provenance of various data sources. Semantic Web could be used to integrate data from various sources and ensure that the sensors were of the same type and context. By using this technique, the current integration process which relies upon administrator physically entering the information could be automated. This would bring our framework one step closer to the autonomous goal desired for IoT.

- Another interesting future work for this framework would be to integrate a data serving layer which would enable the sharing of the various datasets created within the framework. When implementing this framework over a shared configuration, a variety of sensors may be connected to label data and provide analysis. Currently, only the owner of the sensor stream would have access to the labelled data. Although possible, obtaining the labelled data is not designed as a shared functionality of the framework. The ability of the framework to serve labelled data on demand could be an interesting avenue to explore. A repository of all connected sensors and their associated datasets could provide some value to data researchers all over the world.

- There are two important aspects that are not currently being considered in our work: security and privacy. By enabling data analytics through sensor label acquisition, the activities performed by users interacting with sensors may be exposed. Although the users taking part in the framework are aware of those implications, the actions of other non-willing participants may be uncovered. Future works should focus on evaluating how severely the deployment of our framework could affect security and privacy.

Once again, due to the flexibility and modularity of the architecture, a variety of future work could be integrated by either replacing existing components or by simply adding those components to the framework. The design of the framework itself was meant to facilitate the replacement of any components, given that the module still performs the same functionality. The architecture is in no way tied to its implementation.

The work presented in this thesis serves as a proof of concept that gamification can be leveraged as a successful crowdsourcing technique. The future work stresses the importance to integrate concepts that will render the framework more reactive and autonomous because the work presented in this thesis only serves as the stepping stone to enabling autonomous sensor Big Data analytics in the context of the IoT.

Bibliography

[1] I. Cleland, M. Han, C. Nugent, H. Lee, S. McClean, S. Zhang, and S. Lee, "Evaluation of prompted annotation of activity data recorded from a smart phone," *Sensors*, vol. 14, no. 9, p. 15861, 2014.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[3] M. Swan, "Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, pp. 217–253, 2012.

[4] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[5] C. Callison-Burch and M. Dredze, "Creating speech and language data with amazon's mechanical turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 1–12, Association for Computational Linguistics, 2010.

[6] A. Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical turk," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 0, pp. 1–8, 2008.

[7] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vision*, vol. 77, pp. 157–173, May 2008.

[8] Amazon, "Mechanical turk," 2015. https://www.mturk.com/mturk/welcome.

[9] M. Hayes and M. Capretz, "Contextual anomaly detection framework for big sensor data," *Journal of Big Data*, vol. 2, no. 1, 2015.

[10] Y. Liang, X. Zhou, B. Guo, and Z. Yu, “Activity recognition using ubiquitous sensors: An overview,” *Creating Personal, Social, and Urban Awareness through Pervasive Computing*, p. 22, 2013.

[11] S. Deterding, M. Sicart, L. Nacke, K. O’Hara, and D. Dixon, “Gamification. using game-design elements in non-gaming contexts,” in *CHI ’11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’11, (New York, NY, USA), pp. 2425–2428, ACM, 2011.

[12] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner, “mcrowd: A platform for mobile crowdsourcing,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’09, (New York, NY, USA), pp. 347–348, ACM, 2009.

[13] S. Harada, J. Lester, K. Patel, T. S. Saponas, J. Fogarty, J. A. Landay, and J. O. Wobbrock, “Voicelabel: Using speech to label mobile sensor data,” in *Proceedings of the 10th International Conference on Multimodal Interfaces*, ICMI ’08, (New York, NY, USA), pp. 69–76, ACM, 2008.

[14] I. Celino, D. Cerizza, S. Contessa, M. Corubolo, D. Dell’Aglio, E. Valle, and S. Fumeo, “Urbanopoly – a social and location-based game with a purpose to crowdsource your urban data,” in *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pp. 910–913, Sept 2012.

[15] R. Kirkham, C. Shepherd, and T. Plötz, “Blobsnake: Gamification of feature extraction for “plug and play” human activity recognitionplug and play’ human activity recognition,” in *Proceedings of the 2015 British HCI Conference*, British HCI ’15, (New York, NY, USA), pp. 74–81, ACM, 2015.

[16] A. Tarasov, S. J. Delany, and C. Cullen, “Using crowdsourcing for labelling emotional speech assets,” in *Proceedings of W3C workshop on Emotion ML*, Dublin Institute of Technology, October 2010.

[17] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles, “Structured labeling for facilitating concept evolution in machine learning,” in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’14, (New York, NY, USA), pp. 3075–3084, ACM, 2014.

[18] D. Ennis, *Using Pre-and Post-Process Labeling Techniques for Cluster Analysis*. PhD thesis, University of Nevada, Reno, 2014.

[19] O. Alonso, “Challenges with label quality for supervised learning,” *J. Data and Information Quality*, vol. 6, pp. 2:1–2:3, Mar. 2015.

[20] C. Thiel, “Classification on soft labels is robust against label noise,” in *Knowledge-Based Intelligent Information and Engineering Systems* (I. Lovrek, R. Howlett, and L. Jain, eds.), vol. 5177 of *Lecture Notes in Computer Science*, pp. 65–73, Springer Berlin Heidelberg, 2008.

[21] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, pp. 78–87, Oct. 2012.

[22] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, *et al.*, “Constrained k-means clustering with background knowledge,” in *ICML*, vol. 1, pp. 577–584, 2001.

[23] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, “Enhancing data analysis with noise removal,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, pp. 304–319, March 2006.

[24] M. Saraee, N. Ahmadian, and Z. Narimani, “Data mining process using clustering: a survey,” *Proceedings of IDMC’07*, pp. 1–8, 2007.

[25] P. Berkhin, “A survey of clustering data mining techniques,” in *Grouping Multidimensional Data* (J. Kogan, C. Nicholas, and M. Teboulle, eds.), pp. 25–71, Springer Berlin Heidelberg, 2006.

[26] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)19th International Conference in Pattern Recognition (ICPR).

[27] M. Greenacre and R. Primicerio, *Multivariate Analysis of Ecological Data:*. Manuales Fundación BBVA, Fundación BBVA, 2014.

[28] C. Ding and X. He, “K-nearest-neighbor consistency in data clustering: incorporating local information into global optimization,” in *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 584–589, ACM, 2004.

[29] F. Schnizler, T. Liebig, S. Marmor, G. Souto, S. Bothe, and H. Stange, “Heterogeneous stream processing for disaster detection and alarming,” in *Big Data (Big Data), 2014 IEEE International Conference on*, pp. 914–923, Oct 2014.

[30] C. Dule and K. Rajasekharaiah, “Sensor data mining model and system design: A review,” *International Refereed Journal of Engineering and Science*, vol. 2, pp. 16–22, June 2013.

[31] C. Aggarwal, “An introduction to sensor data analytics,” in *Managing and Mining Sensor Data* (C. C. Aggarwal, ed.), pp. 1–8, Springer US, 2013.

[32] N. H. Gehani and H. V. Jagadish, “Composite event specification in active databases: Model and implementation,” pp. 327–338, 1992.

[33] F. Wang, C. Zhou, and Y. Nie, “Event processing in sensor streams,” in *Managing and Mining Sensor Data* (C. C. Aggarwal, ed.), pp. 77–102, Springer US, 2013.

[34] A. Paschke and H. Boley, ch. Rules Capturing Events and Reactivity, pp. 215–252. Hershey, PA, USA: IGI Global, 2009.

[35] W.-K. Wong and D. B. Neill, “Tutorial on event detection kdd 2009,” *Age*, vol. 9, p. 30, 2009.

[36] L. Ye, Z. guang Qin, J. Wang, and J. Jin, “Anomaly event detection in temporal sensor network data of intelligent environments,” in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, vol. 7, pp. V7–414–V7–420, April 2010.

[37] A. Ihler, J. Hutchins, and P. Smyth, “Adaptive event detection with time-varying poisson processes,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’06, (New York, NY, USA), pp. 207–216, ACM, 2006.

[38] C. Piciarelli, C. Micheloni, and G. Foresti, “Trajectory-based anomalous event detection,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, pp. 1544–1554, Nov 2008.

[39] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, “Weakly supervised recognition of daily life activities with wearable sensors,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 2521–2537, Dec 2011.

[40] T. Sztyler, J. Völker, J. Carmona, O. Meier, and H. Stuckenschmidt, “Discovery of personal processes from labeled sensor data - an application of process mining to personalized health care,” in *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, ATAED 2015, Satellite event of the conferences: 36th International Conference on Application and Theory of Petri Nets and Concurrency Petri*

Nets 2015 and 15th International Conference on Application of Concurrency to System Design ACSD 2015, Brussels, Belgium, June 22-23, 2015., pp. 31–46, 2015.

[41] D. Roggen, K. Förster, A. Calatroni, A. Bulling, and G. Tröster, "On the issue of variability in labels and sensor configurations in activity recognition systems," in *Workshop at the 8th International Conference on Pervasive Computing (Pervasive 2010)*, 2010.

[42] I. P. Machado, A. L. Gomes, H. Gamboa, V. Paixao, and R. M. Costa, "Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization," *Information Processing & Management*, vol. 51, no. 2, pp. 204 – 214, 2015.

[43] K. Murao and T. Terada, "Labeling method for acceleration data using an execution sequence of activities," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, (New York, NY, USA), pp. 611–622, ACM, 2013.

[44] T. Miu, T. Plötz, P. Missier, and D. Roggen, "On strategies for budget-based online annotation in human activity recognition," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, (New York, NY, USA), pp. 767–776, ACM, 2014.

[45] L. Zhao, G. Sukthankar, and R. Sukthankar, "Incremental relabeling for active learning with noisy crowdsourced annotations," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pp. 728–733, Oct 2011.

[46] I. Guyon, G. Cawley, G. Dror, and V. Lemaire, "Design and analysis of the wcci 2010 active learning challenge," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1–8, July 2010.

[47] E. Thomaz, T. Plötz, I. Essa, and G. Abowd, "Interactive techniques for labeling activities of daily living to assist machine learning," in *Proceedings of Workshop on Interactive Systems in Healthcare*, November 2011.

[48] C. Aggarwal, "An introduction to data streams," in *Data Streams* (C. Aggarwal, ed.), vol. 31 of *Advances in Database Systems*, pp. 1–8, Springer US, 2007.

[49] M. A. Osborne, S. J. Roberts, A. Rogers, and N. R. Jennings, "Real-time information processing of environmental sensor network data using bayesian gaussian processes," *ACM Trans. Sen. Netw.*, vol. 9, pp. 1:1–1:32, Nov. 2012.

[50] A. Kittur, “Crowdsourcing, collaboration and creativity,” *XRDS*, vol. 17, pp. 22–26, Dec. 2010.

[51] V. S. Sheng, F. Provost, and P. G. Ipeirotis, “Get another label? improving data quality and data mining using multiple, noisy labelers,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’08, (New York, NY, USA), pp. 614–622, ACM, 2008.

[52] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang, “Repeated labeling using multiple noisy labelers,” *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 402–441, 2014.

[53] M. R. Smith and T. Martinez, “Becoming more robust to label noise with classifier diversity,” *arXiv preprint arXiv:1403.1893*, 2014.

[54] V. C. Raykar and S. Yu, “Eliminating spammers and ranking annotators for crowdsourced labeling tasks,” *J. Mach. Learn. Res.*, vol. 13, pp. 491–518, Feb. 2012.

[55] C. Thiel, “Classification on soft labels is robust against label noise,” in *Knowledge-Based Intelligent Information and Engineering Systems* (I. Lovrek, R. Howlett, and L. Jain, eds.), vol. 5177 of *Lecture Notes in Computer Science*, pp. 65–73, Springer Berlin Heidelberg, 2008.

[56] M. Buhrmester, T. Kwang, and S. D. Gosling, “Amazon’s mechanical turk: A new source of inexpensive, yet high-quality, data?,” *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.

[57] D. Yang, G. Xue, G. Fang, and J. Tang, “Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones,” *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–13, 2015.

[58] R. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *Communications Magazine, IEEE*, vol. 49, pp. 32–39, November 2011.

[59] S. Kanhere, “Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces,” in *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, vol. 2, pp. 3–6, June 2011.

[60] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, “Fostering participaction in smart cities: a geo-social crowdsensing platform,” *Communications Magazine, IEEE*, vol. 51, pp. 112–119, June 2013.

[61] P.-H. Tsai, Y.-J. Lin, Y.-Z. Ou, E.-H. Chu, and J. Liu, "A framework for fusion of human sensor and physical sensor data," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, pp. 1248–1261, Sept 2014.

[62] "What is gamification?," tech. rep., Gamification Summit, 2013.

[63] S. Nicholson, "A user-centered theoretical framework for meaningful gamification," *Games+ Learning+ Society*, vol. 8, no. 1, 2012.

[64] C. E. Bess, "Gamification: Driving behavior change in the connected world," *Cutter IT Journal*, vol. 26, no. 2, pp. 31–37, 2013.

[65] J. Cechanowicz, C. Gutwin, B. Brownell, and L. Goodfellow, "Effects of gamification on participation and data quality in a real-world market research domain," in *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, Gamification '13, (New York, NY, USA), pp. 58–65, ACM, 2013.

[66] G. Kazai, Lumi, F. Hopfgartner, U. Kruschwitz, and M. Meder, "Ecir 2015 workshop on gamification for information retrieval (gamifir'15)," *SIGIR Forum*, vol. 49, pp. 41–49, June 2015.

[67] N. Cameron, "Why gamification and big data go hand-in-hand," tech. rep., CMO: Marketing, Technology, Leadership, September 2013.

[68] K. Han, E. Graham, D. Vassallo, and D. Estrin, "Enhancing motivation in a mobile participatory sensing project through gaming," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pp. 1443–1448, Oct 2011.

[69] L. Barrington, D. Turnbull, and G. Lanckriet, "Game-powered machine learning," *Proc Natl Acad Sci U S A*, vol. 109, pp. 6411–6416, Apr 2012. 22460786[pmid].

[70] J. M. Gomes, T. Chambel, and T. Langlois, "Soundslike: Movies soundtrack browsing and labeling based on relevance feedback and gamification," in *Proceedings of the 11th European Conference on Interactive TV and Video*, EuroITV '13, (New York, NY, USA), pp. 59–62, ACM, 2013.

[71] K. Dergousoff and R. L. Mandryk, "Mobile gamification for crowdsourcing data collection: Leveraging the freemium model," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), pp. 1065–1074, ACM, 2015.

[72] J. He, M. Bron, L. Azzopardi, and A. de Vries, "Studying user browsing behavior through gamified search tasks," in *Proceedings of the First International Workshop on Gamification for Information Retrieval*, GamifIR '14, (New York, NY, USA), pp. 49–52, ACM, 2014.

[73] B. Ostermaier, K. Römer, F. Mattern, M. Fahrmair, and W. Kellerer, "A real-time search engine for the web of things," in *Internet of Things (IOT), 2010*, pp. 1–8, IEEE, Nov 2010.

[74] Parse, "Parse," 2015. https://www.parse.com.

[75] Oracle, "Java," 2015. https://www.java.com.

[76] M. Oderski *et al.*, "An Overview of the Scala Programming Language," No. IC/2004/64, 2004.

[77] Play, "Play framework," 2015. https://www.playframework.com.

[78] M. Caglioti, M. Detriech, H. Emonds, and L. Smith, "Software capstone project," May 2014.

[79] R Core Team, *R: A Language and Environment for Statistical Computing*. 2013.

[80] RStudio, Inc, *shiny: Easy web applications in R*, 2014.

[81] Powersmiths, "Powersmiths: Power for the future," 2010. http://ww2.powersmiths.com.

[82] SmartThings, "Smartthings," 2015. http://www.smartthings.com.

[83] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[84] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Greenwich, CT, USA: Manning Publications Co., 1st ed., 2015.

Curriculum Vitae

Name:	Alexandra L'Heureux
Year of Birth:	1988
Place of Birth:	Sherbrooke, Quebec, Canada
Post-Secondary Education and Degrees:	Western University London, ON 2013-2015 MESc.
	Western University London, ON 2009 - 2013 BESc.
Honours and Awards:	NSERC CGS M 2014-2015
	Ontario Graduate Scholarship 2013-2014
	Best 4th year design project 2013
	Christian Lassonde Scholarship 2012
	NSERC USRA 2012

Related Work Experience:	Teaching Assistant Western University 2013 - 2015
	Summer Research Intern Powersmiths International 2014
	Designer and Developer Lawson Health Research Institute 2012 - 2013

Publications:

- A. Tiwari, A. Haji, **A. L'Heureux**, W. Hunt, and M. Capretz, R. Mann, "*Biometrics in the Mental Health Community*" International Conference for Upcoming Engineers, 2012.

- K. Grolinger, M. Hayes, W.A. Higashino, **A. L'Heureux**, D.S. Allison, M. Capretz, "*Challenges for MapReduce in Big Data*" Proceedings of the IEEE 10th 2014 World Congress on Services (SERVICES 2014), 2014.