MULTI PERSON TRACKING AND QUERYING

WITH HETEROGENEOUS SENSORS

by

SHAHINA FERDOUS

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2012

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. Fillia Makedon and Dr. Leonidas Fegaras for their constant support and guidance throughout my entire PhD study. I would also like to sincerely thank my committee members, Dr. Bahram Khalili and Dr. Chengkai Li for their input and cooperation.

I am very thankful to all the members of the Heracleia Lab. Without their ideas and participation, my research would not been possible. Finally, special thanks go to my husband and my parents, who were always there for me and gave me the inspiration to continue my career.

November 26, 2012

ABSTRACT

MULTI PERSON TRACKING AND QUERYING
WITH HETEROGENEOUS SENSORS

SHAHINA FERDOUS, PhD

The University of Texas at Arlington, 2012

Supervising Professor: Fillia Makedon and Leonidas Fegaras

Tracking the location of a user is considered to be the most fundamental step for creating a context aware application, such as activity monitoring in an assistive environment. This problem becomes very challenging if there are multiple people involved in this scenario. The reason is that any multi-person environment, such as a hospital, demands simultaneous identification and localization mechanisms, thus making the system very complex. In this dissertation, we present a novel, less-intrusive system that uses RFIDs and sensors deployed at various locations of an assistive apartment to continuously track and identify every person in a multi-person assistive environment. Our experimental result proves the prospect of using RFID and sensors jointly to solve the simultaneous tracking and identification problem. In addition, this system stores the large scale spatio-temporal sensor data into a common repository and provides a flexible query interface to track the history of the patient. The visualization tool embedded in the system helps therapists to remotely monitor a person present in a scene in near real time. Such a visualization gives a very good indication about a person/patient's activity and behavior in the assistive environment. Our system also

incorporates the metadata mapping of the large amount of stored data so that a doctor/therapist can query about a patient's records without having a complete knowledge about the schemas stored in the repository.

TABLE OF CONTENTS

Chapter Page

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Problem

An assistive environment is defined as an area that supports people living or working in that space through its embedded technologies. With the recent advancement in assistive technologies, a large number of applications have emerged to support the healthcare industry. Among them, the most commonly reported applications are monitoring elderly people at home or patients in a hospital, tracking assets, such as drugs or inventory, building smart furniture and developing reminder systems etc. [1, 2, 3, 4, 5]. Automatic collection and transfer of data to analyze and track history is another key area of interest for such applications. As a whole, the purpose of healthcare applications can be divided into two main directions: 1) real time monitoring and assistance, time driven reminder of daily activities, and generating alarms or notifying a responsible person in case of emergency; 2) organizing and analyzing statistical data in order to identify or query useful patterns related to a particular patient or a group of patients.

Recent advancement in pervasive technology makes the automated monitoring of elderly people living alone in a home much easier. Different pervasive sensors such as IR, motion sensors, pressure pads and sunspots can be easily integrated to the daily living spaces without making the system invasive to the privacy of its residents. Although such passive sensors are sufficient to summarize activities in an assistive environment, where an elderly lives alone, they are not sufficient for assistive environments with many people, such as hospitals. But, current assistive environments should be smart enough to monitor the interaction among multiple humans or between a human and computer in real time. For an assistive environment with many people, identifying and tracking the locations of every person simultaneously is considered to be one of the biggest challenges. In addition, data generated from such pervasive applications are of large volume and can be of many different types or schemas, such as sensor readings, text, audio, video, medical records, etc. But an effective application needs to store such a huge variety of data into a common repository and provide an easier access to that repository. A caregiver or doctor needs to query over that repository to track the status and history of a patient. But it is very hard for them to understand and remember each such different schema to query the data repository. It is also highly infeasible for a system to cache all possible answers to a query beforehand, as the number of queries can vary a lot depending on user's requirements. As a result, pervasive applications should incorporate an *interoperability standard* [3] and metadata mapping techniques such that caregivers or clinicians can query about a patient conveniently, irrespective of the schemas or the types of data stored on the repository.

1.2 Motivation

Automatic identification and tracking is very useful for applications [6, 7], such as surveillance, monitoring movements and activities of assistive daily living (ADL). For example, automatic identification and localization methods are key to detect intruders or to track suspicious movements. Similarly, remote monitoring of assistive daily living requires continuous location tracking of every person in order to measure movements, identify activities, predict or detect accidents, falls, or other emergency situations, and summarize behaviors. As a result, a system is needed that visualizes the status of every person present in a multi-person environment in order to help the caregivers or families to monitor them remotely. This system must also integrate a flexible query interface, understandable and usable by any human, who acquitted the proper authority. Such a query interface would help the researchers and caregivers to analyze and track history and predict behaviors and wellness of the patients.

Studies show that video cameras are used most commonly in current ADL to track multiple people, as cameras not only help in localizing, but also capture information about a person's pose and interaction with other objects or people in the vicinity. However, capturing this information using only cameras is very challenging for a wide-space containing many people as identification requires zooming into a person's face, while localization requires wide coverage to capture and map the respective locations of many people simultaneously. This problem becomes harder, if we consider long periods of time and large number of people involved.

With the recent advancement in the mobile phone technology, most of the mobile phones contain embedded accelerometers and magnetometers. As a result, methods that combine localization using a camera with identification using wearable sensors or accelerometers are also proposed in literature to solve the problem of tracking people. But for an unconstrained environment with multiple people, information collected through passive sensors may found to be very noisy and thus, could lead to incorrect identification [3]. For a very dynamic environment, information collected from multiple sources, such as video cameras, microphone array, sensors etc. are all combined together, such that the system can achieve better identification accuracy. Although the use of cameras and computer vision techniques are very promising, current methods do not completely solve the problem of identifying and localizing multiple persons, while extensive use of video cameras in an assistive applications are invasive to person's privacy [8]. Therefore, our main focus is to develop a system that can achieve the same goal of identifying and localizing multi-persons in an assistive environment using sensors in a less-intrusive manner.

1.3 Contributions

In this work, we present a PeopleTrack System that 1) visualizes the locations of every person present in an assistive environment, 2) collects and stores the spatio-temporal data in a common repository and 3) provides a fast and flexible query interface, such that the clinicians or caregivers can query about any information without even knowing the schema that represents the data. We have used various less-intrusive sensor technologies and data mining approaches to solve the identification and localization problem. In addition, we have applied state of the art data mining algorithms to analyze the data in order to identify activities and predict behaviors. The system we present helps the caregivers or doctors to monitor a patient in near real time, as wells as helps them to query about history of patient records or the level of activities. However, to make the querying faster and easier, we have applied suitable data mining techniques to model and map metadata of the large amount of data stored in the repository. Although our system is mainly designed to monitor and track assistive daily living, such a system can also be used in other applications, such as Warehouses, large Hospitals. The rest of this thesis is organized as follows: chapter 2 summarizes the related work; chapter 3 describes a tool that simulates computer aided sensor placement and activities in an assistive apartment; chapter 4 describes collection of different types of sensor data required for our system; chapter 5 describes simultaneous identification and localization of every people present in a scene; chapter 6 presents a web based tool to monitor medication intake patterns; chapter 7 describes automatic assessment of depression from text records; chapter 8 describes modeling large-scale sensor data to provide faster responses to historical queries; chapter 9 describes a framework to integrate and query different types of data generated from pervasive applications and chapter 10 summarizes the conclusion.

CHAPTER 2

RELATED WORK

2.1 Introduction

Multimodal person identification has become a significant area of interest in recent pervasive assistive applications. Some of these applications use existing biometric identification methods, such as face recognition and speaker identification [9, 4], in order to identify multiple people in smart environments [10]. Although these works focus on person identification in a multi-person environment, they do not convey the location information of the person completely. But, locating multiple people simultaneously while identifying them is considered to be the first step for creating context-aware applications, such as activity monitoring and human behavior recognition in an assistive environment. The next few subsections summarize all different directions that the researchers have followed to solve this problem [11].

2.1.1 Approaches based on Video Cameras

Some researchers have used discriminative appearance based affinity models to identify and track multiple people in a complex scene with a single camera [12]. Several other robust multi-person tracking approaches based on tracking-by-detection have also been discussed in literature [13, 14, 15, 16, 17]. In [18], the authors address the problem of multi-person tracking by using a single mobile camera mounted on a vehicle or mobile robot. They propose a robust image level tracker based on level-set segmentation that tracks each pedestrian using an automatically initialized level-set.

To solve person tracking in a real-world scenario, some researchers have also used images from two colored stereo cameras that help to identify and locate multiple people [19]. The cameras they use maintain color histograms of the person-shaped blob to disambiguate between people who are very close to each other. Their system tracks multiple people standing, walking, sitting, occluding, entering and leaving, near real-time. Multiple camera-based 3-D tracking approaches are also proposed in literature [20]. In [20], the authors considered two tracking approaches: 1) best-hypothesis heuristic tracking and 2) probabilistic multi-hypothesis tracking to derive the 3-D locations of people. Their results show similar tracking performance for both approaches. However, the simplistic probabilistic approach produces more false alarms, which may be compensated by using a sophisticated probabilistic model, which is left for future work.

2.1.2 Approaches based on RFID

Although radio signal propagation suffers various problems, such as multipath, line of sight path, diffraction or reflection, even in an indoor environment [21], several indoor-based localization algorithms have been proposed in literature to achieve better localization accuracy. The methods can be classified into three categories: 1) distance estimation, 2) scene analysis and 3) proximity [22]. Among them, distance estimation algorithms use different range measurement techniques, such as Received Signal Strength, Time of Arrival, Time Difference of Arrival, Received Signal Phase, and apply triangulation to estimate a location. On the other hand, the scene analysis approaches first measure fingerprints (information) of an environment and then try to match a target’s range measurements with the appropriate set of fingerprints for estimating the location. However, the proximity-based algorithms determine a target’s location by mapping it to the location of an antenna that receives the strongest signal.

Overall, RFID technology posses promising solution to identify and localize multiple objects with attached RFID tags [23, 24]. Existing well-known systems, such as SpotON [25] and LANDMARC [26], use active RFID tags and exploit signal strength property to correctly localize an object. Passive RFID tags have also been applied in the past to identify and locate multiple objects [23, 24, 25]. In [23], the authors have utilized the percentage of tag counts at different power attenuation levels in order to approximate the distance between a reader and a tagged object. Research also suggests that RFID technology can be utilized to extract both identification and location information of moving objects [19, 20]. According to these studies, a mobile robot equipped with two RFID antennas can be used to learn a sensor model that represents the likelihood of reading a tag, given the relative position between the tag and the antenna is known a priori. Such a sensor model can be used to predict the location/position of the moving objects or person from the sample of detected tags during the test phase.

2.1.3 Approaches based on Multiple Modalities

The key research goal of a pervasive assistive environment is to develop an integrated sensor-camera-microphone-based system that is autonomous and efficient enough to monitor every human-computer interaction happening over time, in spaces that are not just limited to indoor environment [27]. The following sub-sections describe different previous works based on multiple modalities.

2.1.3.1 Using Cameras and Microphones

Research suggests that arrays of cameras and microphones embedded in an assistive environment can extract effective features about different events and activities [15, 27]. State-of-the-art classification and clustering techniques, such as the Hidden Markov Model, the K-nearest neighbors, can be applied on the captured audio-video signals to extract higher-level semantic information, such as identification and location etc., in real time. However, an integrated system that combines speech, lip-motion and face images, has also been proposed in literature to improve identification accuracy [28].

With the recent trend of mobile devices being ubiquitous and more computationally powerful, researchers are also studying face and speaker identification techniques applicable to mobile devices [29]. A system that combines face and audio-based identification along with motion detection, person tracking and audio based localization, has also been proposed in literature [30]. Such a system applies state-of-the-art novel methods to process results from individual modality, while uses particle filtering to fuse both modalities for providing robust identification and localization.

2.1.3.2 Using Cameras and Accelerometers

Tracks from wearable accelerometers and cameras can be combined together to identify and localize multiple people in a scene [16, 17, 31]. According to this technique, the motion signatures about a person's gait are detected for each person in a scene first. Next, the motion signatures from the accelerometer worn by the respective person are captured too. Finally, the system applies clustering to detect the matching between camera and accelerometer track signals, which derives location information. Since each accelerometer is associated with a unique ID, it makes the identification much easier. The same authors also proposed the use of an existing CCTV infrastructure based system along with sensors (accelerometers and magnetometers) embedded to a person's mobile phones to solve the same problem [6]. According to this method, the camera captures the location of each person, which is transmitted wirelessly to the mobile devices carried by every person. After receiving the location information, the mobile phones resolve the most probable location for them by matching those with the measurements from its own sensors. The identification process is easy in this case too, as each person is labeled with his/her mobile phone's unique ID.

2.1.3.3 Using RFID and Cameras / Passive Sensors

The deployment of wireless sensor network (WSN) is another common approach now-days to monitor and localize person in assistive environments [32, 33]. As identification with RFID is near accurate, an RFID system and a WSN can be combined together not only for identifying and localizing objects, but also for real-time monitoring in assistive environments [34, 35]. Researchers in [36] calculate the location uncertainties of each person from received RFID signal strength and apply the nearest neighbor algorithm from the point of sensor activations to resolve the uncertainties.

Camera and RFIDs are used jointly in literature to achieve sufficient accuracy for localization as well. To identify and localize in open areas, some researchers derived a calibration method for joint RFID-camera system based on the area of overlap between field of view (FoV) of a camera and field of sense (FoS) of RFID sensors [2].

2.1.4 Querying over Large Scale Sensor Data

To answer high-level queries over the large volumes of continuously changing spatio-temporal sensor Data, significant amount of processing time and storage capacity are required, if no optimization techniques have been adopted. Earlier work [37] describes a Data Warehousing technique to compress and aggregate massive RFID Data so that various high-level queries can be answered efficiently over this Data. The main assumption of their proposed architecture is that most of the RFID objects tend to stay and move together. They achieve significant reduction over their Data by compressing multiple rows for objects that move to the same location into a single row instead of adding an individual row for each object.

In healthcare system, tracking of distributed patient-drug interactions is very important to answer questions, such as, which drugs are consumed by whom, when and where. The Bonsai Development Corporation (BDC) incorporates distributed, low-cost device technology to store and aggregate large volumes of Patient-Drug interactions data captured through tracking the distribution of drugs to the patients across a wide geographical area [38]. A centralized Data center stores information from all the distributed sources and can answer queries, such as tracking of locations and drug history of Patients and tracking of drug distribution both in batch and serialized levels.

Earlier work [39] describes a temporal RFID data model based on the concept of the dynamic relationship ER model [DRER]. The proposed Data model supports evaluation of complex queries by capturing continuous state changes and tracking history of events. In this paper, the author identifies fundamental RFID entities, which is either static or dynamic and defines dynamic relationships and interactions among those entities over time. The system transforms fundamental RFID logics, observations and events into high-level semantic Data as well as into more complex business logics using a Rule based framework. The framework applies Data Filtering, Location Transformation and Data Aggregation techniques to perform the transformation.

CHAPTER 3

COMPUTER-AIDED SENSOR PLACEMENT

3.1 Introduction

Placing sensors of interest at important locations of an assistive apartment is the key to provide maximum coverage that makes the remote monitoring and activity recognition more accurate. Applications in assistive environments generally use Finite State Machines (FSM) to recognize such human activity [40][41][42]. But, construction of these FSM strictly depends on the number and type of available sensors as well the location of these sensors in an assistive environment. However, given a fixed layout of an apartment along with the location and type of sensors placed in it, modeling the FSM is straightforward.

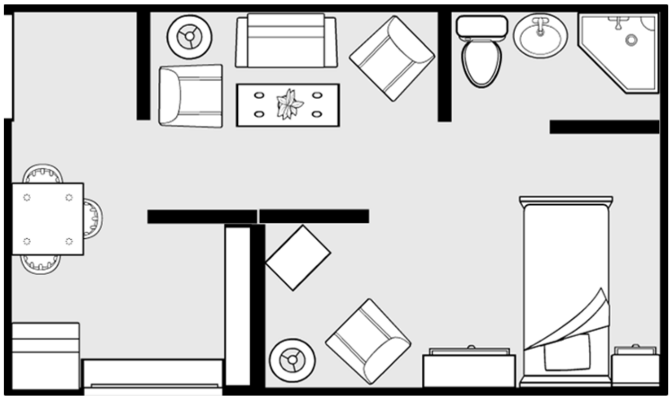


Figure 3.1 A sample apartment layout.

Therefore, our system includes a graphical tool [43] that automatically generates a FSM for an assistive living environment. Figure 3.1 shows an example layout of an assistive living apartment. According to our design of the graphical interface, a user (researcher / technician) can choose one or more sensors from a list of available sensors provided in the tool and can place these sensors at different locations. Depending on the number of selected sensors, their type and area of coverage, the tool automatically generates an FSM for that particular setup. The tool also simulates human actions in that setup and uses the FSM to determine the position of a person in the assistive living environment.

3.2 The Graphical Tool

According to the design, we define a particular region as critical area (shown in blue in Figure 3.2), if a person generally stays longer in that region during his/her daily living. Example of such critical areas include: living room, kitchen, bed, bathroom, etc. As a result, assuming that a researcher / technician provides the layout of the apartment and defines the critical areas, and the type and position of the sensors, the tool will automatically generate a Finite State Machine that models human activity in that apartment. However, the tool does not impose any restriction on the number of users living in the apartment, since that depends on how the FSM is processed later.

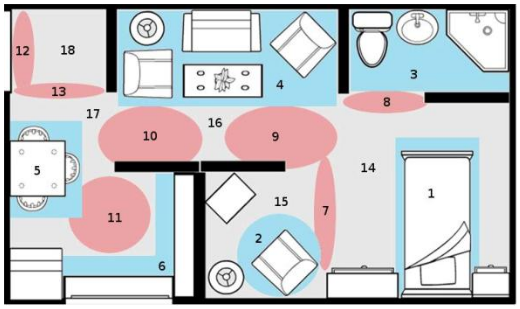


Figure 3.2 The blue areas are the critical areas, the red areas are the areas covered by the sensors and the gray areas are the temporary states

3.2.1 Description of the Graphical Tool Interface

Provided that a researcher/technician first upload a layout of the environment, the graphical tool suggests a collection of sensor types that the researchers (or technician) can choose from. The user can also define the sensor properties, such as range and sampling frequencies. Finally, the user can simply drag and drop the sensors to place them at location of interest.

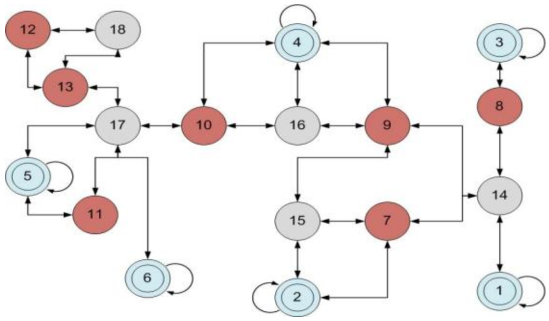


Figure 3.3 The resulting Finite State Machine.

3.2.2 Transition to FSM

Our system generates the Finite State Machine as follows:

1. It assumes that there exists a starting state, where previous knowledge of the user position is unknown. All critical areas are represented as final states, since they represent the areas where a user is likely to remain for a long, unpredictable period of time.
2. All other areas covered by sensors, where a user is not likely to remain for a long period of time are represented as non-final states. The tool knows with great certainty when a user is in that area (assuming non faulty sensors).
3. However, a non-critical area that is not covered by a sensor is also represented as a non-final state, since it is an area where a user is not likely (or it is not reasonable for the user) to remain for a long period of time (eg. a corridor).
4. A sensor activation changes the current state to a state that represents the area covered by that activated sensor.
5. Sometimes, a sensor inactivity may also trigger an alarm and/or a state transition from a non-final state to another.

Non-final states may actually be represented by more than one state. Non-final states may contain information about the previous position of the simulated person that is used to estimate the next state, or may represent that the person is in specific areas with certain probability. The non-final states generated from not covered areas are the ones, for which the tool is the least confident, since it can only determine that a person is there by inference. For example, in Figure 3.3, if a user is in state 10 and the corresponding sensor stops detecting the user, he could be either in state 17 or in state 16. As a result, our system has to wait for another sensor activation to determine the user's actual position.

3.2.3 Discussions

Such a tool helps a lot to simulate a person living in an assistive living environment. The researcher / technician can click a sensor or a series of sensors, simulating sensor activations, and then see the estimated user position. The tool also provides warnings for uncovered areas or ambiguities. The simulation results from the tool provide important suggestions concerning the optimal placement of the sensors that the user has selected. Thus, the tool helps the user to design the optimal placement of the sensors, before they deploy them in a real environment which is a time and cost sensitive operation and start collecting Data.

CHAPTER 4

COLLECTION OF LARGE-SCALE SENSOR DATA

4.1 Introduction

Once sensors have been placed optimally in an assistive environment, the next task is to collect and store the raw sensor data or the sensor-events of interest. The sensors that we use mainly for our system are: 1) RFID readers and tags 2) passive sensors, such as IR, motion sensors, pressure mats and sunspots, and 3) Kinect sensors. Since any of these sensors may generate large amount of spatio-temporal data within a very short period, our system incorporates mechanisms to collect and store only the important/meaningful events, instead of the raw sensor data. The subsequent sections in this chapter describes the collection of different types of sensor data based on their applications in our system.

4.2 Collection of RFID Data to Track History of Patient-Drug Interaction

4.2.1 RFID Technology

RFID systems use radio frequency waves to identify physical objects. An RFID System consists of several components, named Tags, Readers, and Application software [44, 45]. Whenever a tag receives electromagnetic radio-frequency signal from a reader, the tag transmits its identifier and data back to the reader. The reader translates these received radio frequency signals into a digital form and sends those data back to a host computer connected to the reader. However, RFID tags can be of active or passive type. Active tags operate at active high frequency, which can be 455 MHz, 2.45 GHz or 5.8 GHz. On the other hand, passive tags operate at frequencies of the range 128 KHz, 13.56 MHz, 850-950 MHz or 2.45 GHz [46]. The suitability of a frequency for a particular application completely depends on the following factors: distance from which a tag can be detected by a reader, permeability (ability of correct read in case of noise) and data transfer rate [47]. As active tags have their own battery power source, they can be read from distances over 100 ft. On the other hand, passive tags power themselves from the electromagnetic waves of the external readers and can be read from distances 20ft at most. With the recent development in this field, the costs for RFID tags and readers are decreasing very rapidly. As a result, this technology is enabling each individual object to be tagged in order to achieve better precision [48]. In fact, both active and passive RFID tags are suitable for our system to track the patient-drug interaction in a large assistive environment such as a hospital. In addition, both fixed and mobile RFID readers can be used as described in [45]. Passive tags can be used to track and identify objects, such as drugs, food, equipment or furniture, which have low mobility or no mobility at all. In that case, fixed RFID readers are deployed at various places of the assistive environment to track mobile objects that change their locations less frequently. As stationary objects, such as large hospital equipment or furniture, such as beds, are not moved very often, in order to track the interaction of a patient with those objects, mobile RFID readers can be used. In addition, a person can carry RFID Reader attached to his/her wristband or to ID cards in order to detect the presence of any stationary objects close to him/her. Moreover, as persons, such as patients, doctors or staff change their locations very frequently, RFID tags can be attached to their wristbands or ID cards to track and monitor them. Since active tags can operate from a reasonable amount of distance, hence deploying smaller number of fixed RFID readers can cover the whole area of such assistive environments.

4.2.2 Description of Data

Data generated from an RFID Reader can be described as a continuous stream of tuples, where each tuple takes the simple form (EPC, Location, Time) [49], as shown in Table 4.1. Here, EPC stands for electronic product code and is used as a standard way to identify an object universally. Location is actually the location of the reader, but it can be used indirectly to refer to the location of an object at the time when the reader detects it.

Table 4.1 Example RFID Data Set

EPC	Time	Location
E2009037890401091080A8BA	T1	L1
E20090378904010910803DGB	T1	L1
E2009037890401091080A8BA	T8	L1
E20090378904010910803DGB	T10	L1
E2009037890401091080A8BA	T11	L1
E20090378904010910803DGB	T11	L2
E2009037890401091080A888	T12	L2
E20090378904010910803DGB	T13	L2
E2009037890401091080A888	T15	L2
E2009037890401091080A8BA	T15	L3
E2009037890401091080A8BA	T20	L3

According to Venture Development Corporation, retail stores, such as Walmart, generate around 7 terabytes of data every day if the tags are placed at item level. Therefore, a simple RFID application can generate a large amount of spatio-temporal data within a very short period. In the context of healthcare systems, each individual patient can ware RFID readers attached to his or her wristbands, while individual items, such as medicine, food, equipment, may have attached RFID tags. If a hospital authority wants to ask queries, such as what are the most frequent drugs according to their average daily/monthly intake, new data structures and algorithms need to be modeled in order to provide fast query responses from these large volumes of dynamically changing, time-sensitive and location-specific data [50].

4.3 Collection of Medicine Intake Data

4.3.1 Physical Setup

Our system includes a smart medicine tracking tool, named Smart Drawer that remotely monitors the daily medication intake of a patient living alone in an assistive apartment [51]. The Smart Drawer has various real-world components, which is actually a part of our simulated apartment as shown in Figure 4.1. The drawer to the right of the bed, which is encircled in red, contains the equipment used to measure the medication.

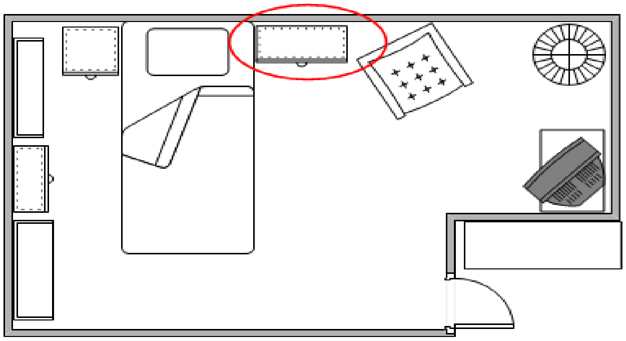


Figure 4.1 Example apartment layout with a Smart Drawer containing RFID reader and other sensor

An RFID reader is placed inside that drawer, which can read values from individual medicine bottles placed on top of it [52]. In addition, a precision balance (an analytic scale) [53], which has a resolution of 0.001 g and a serial connection to report a stream of the current weight of the medicine contents, is also placed in the drawer. Hence, the overall setup inside the drawer contains all the medicine bottles placed on top of a RFID reader, which is eventually put on top of the precision balance. Furthermore, a Sunspot sensor is mounted to the front of the drawer to detect the position of the drawer (either open/closed). In addition, a Motion Detection sensor, from Phidget [54], is mounted on the leg of the lounge chair to detect when a person enters the bedroom. The setup of sensors and spaces mentioned above defines the overall workspace for the Smart Drawer that is necessary to track the medicine intake.

4.3.2 Drawer Events

One of the key sensors of the Smart Drawer system, which is being accessed by the web tool, is Sunspot wireless sensor mote. It has an embedded accelerometer with three axes. Given that the sunspot is mounted on the front of the drawer, the system applies two different methods to determine when the drawer is in use, i.e. either open or shut. We call such detection of drawer motion as drawer events.

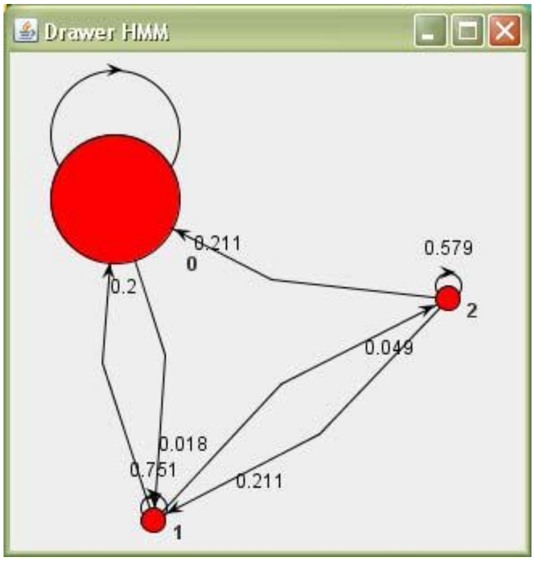


Figure 4.2 Drawer Event HMM

According to the first method, the system detects a drawer event by simply comparing the drawer motion with respect to a predefined threshold.

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

Here, x_j is the current reading, *xi* are the past n readings, t_k is the threshold in gravities for the acceleration of the drawer (0.064 g) and *n* is 30. Hence, when the motion of the drawer exceeds the threshold at certain point, the program on the Sunspot detects an event and sends that signal over the wireless sensor network (WSN).

In the second approach, the same data from sunspot imported to the Java Hidden Markov Model API [55] that allows the data to be processed. By using the K-Means Learner, the status of the drawer can be modeled as shown in Figure 4.2. Once a model is created, the system is programmed such that it recognizes the incoming data streams as observations and can estimate the current state of the drawer. If the system detects the accelerometer to be moving outwards, i.e. moving in the positive *Z* direction, then it concludes that the drawer is open. However, once an event is detected, the Sunspot reports that event either through the wireless sensor network or through a USB cable connected to the central controller of the medicine drawer.

4.4 Collection of Multi-Person Identification and Location Data

4.4.1 Kinect Data Collection

We have used Microsoft Kinect, as shown in Figure 4.3 to collect the location tracking information of multiple person present in a scene [56]. Note that Kinect is a motion-sensing device for the XBOX 360 video game console and Windows PC.

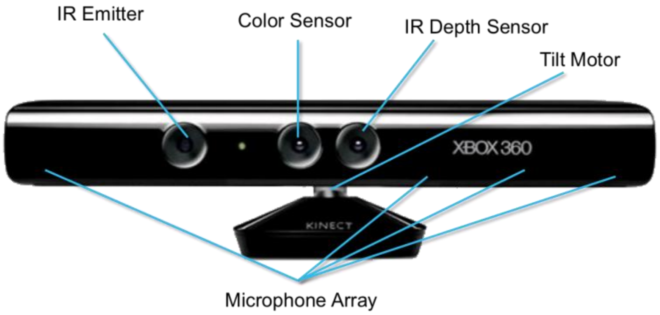


Figure 4.3 Kinect Sensor

However, to integrate a Kinect sensor to our system, we have used the commercially available Kinect SDK for C#. The SDK supports both depth and RGB images, tilt, microphone array and skeletal tracking that gives skeleton data values (shown in Fig. 4.4) in meters with respect to a full 3D coordinate system, as shown in Fig. 4.5.

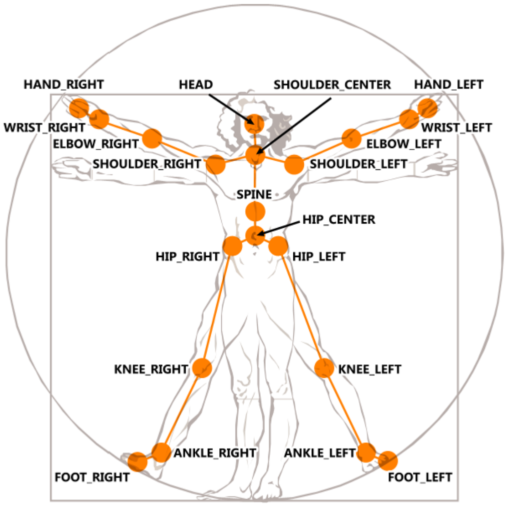


Figure 4.4 Skeleton Data

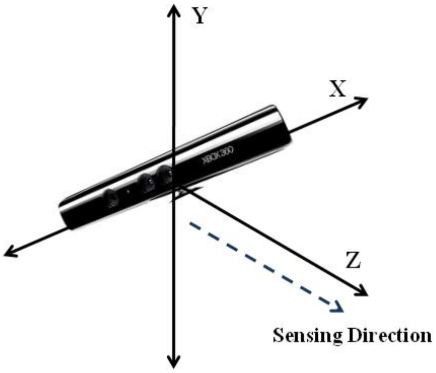


Figure 4.5 Coordinate System for the Skeleton Data

Each skeleton has a unique identifier for a particular session and is defined by the coordinates <x, y, z> of the joints expressed in meters. Each joint can be at any of the three associated states: 1) tracked, 2) not tracked, and 3) inferred. The SDK also allows choosing which skeleton to track at a particular moment.

4.4.2 RFID Data Collection

The RFID system we use to identify multiple persons in a scene is the commercially available Alien 9900+ developer kit. The kit includes a reader with two circularly polarized antennas. The tags used in our experiment are EPC Class 1 Generation 2 supported by the 9900 readers. Figure 4.6 shows an example tag and antenna design from Alien. As the antennas are circularly polarized, the tag orientation is not an issue for our experiment. However, for an indoor environment, the antenna read range for the passive RFID tags varies from 6 to10 ft. Such a read range is sufficient to detect the presence of a person carrying a tag in the simulated rooms of our Heracleia Assistive Apartment, given the tags are within the Field of Sense (FOS) of the antennas.

However, in our experimental setup, we have deployed two antennas at the two corners of the bedroom. We have simulated an experiment for identifying and localizing 4 persons, limited only to that room, although the system can be extended to more people by adding additional Kinect sensors in the apartment. During the experiment, the persons wear the RFID tags around their neck like a normal ID card.

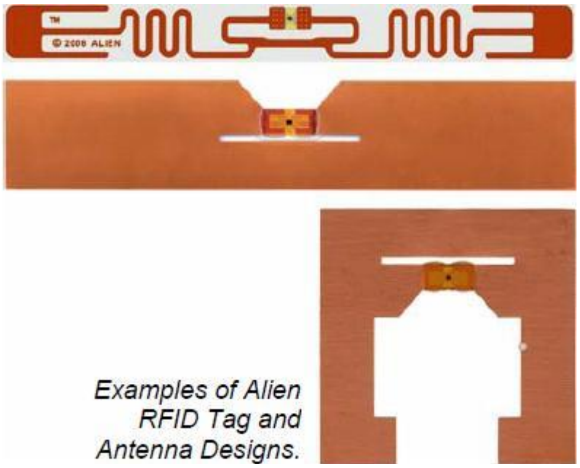


Figure 4.6 Example Alien RFID Tag and Antenna Designs

CHAPTER 5

PERSON IDENTIFICATION AND LOCALIZATION

Tracking the location of a person is considered to be the most fundamental step for creating a context aware application, such as activity monitoring in an assistive environment. But, the problem becomes very challenging if multiple people are involved in this scenario. The reason is that any multi-person environment, such as a hospital, demands a simultaneous identification and localization mechanism, thus making the system very complex. To solve this problem, our system applies a novel, less-intrusive approach that uses RFID and passive sensors or a Kinect sensor deployed at various locations of an assistive environment. Basically, the system maps the RFID events together with the passive sensor activations / kinect tracking information to track the location of every person in a room. However, to map the identification events from the RFID to the location information from the Kinect sensor, the system applies two approaches: 1) a proximity-based and 2) a classification-based. Our evaluation of these two approaches proves their effectiveness in real world scenarios.

5.1 Example Scenario

According to the apartment layout described in Figure 5.1, given readings from a RFID antenna, Ant 0, our system can deduce 3 persons, such as A, B and C, to be present in the room, Room 0. But, in order to deduce the location of each person to a specific level, such as person A is near the door, B is near Bed and C is on the couch, the system needs to consider the location uncertainty of each of the moving objects and relate these uncertainties with respect to different sensor activations [57]. For example, given the current locations of two persons $P1$ and $P2$ near sensor points $S1$ and $S2$ at a moment t, as shown in Figure 5.2, if a bedside sensor $S5$ near point $q1$ activates on (t+1)-th moment, the system needs to calculate the distance between the point $q1$ and the probable locations of each person at time (t+1) in order to identify the nearest person, who activated the sensor. Hence, depending on the speed and direction of each person during [t+1], either $P2$ or $P1$ could be the nearest to $q1$ as shown in Figure 5.2b and 5.2c.

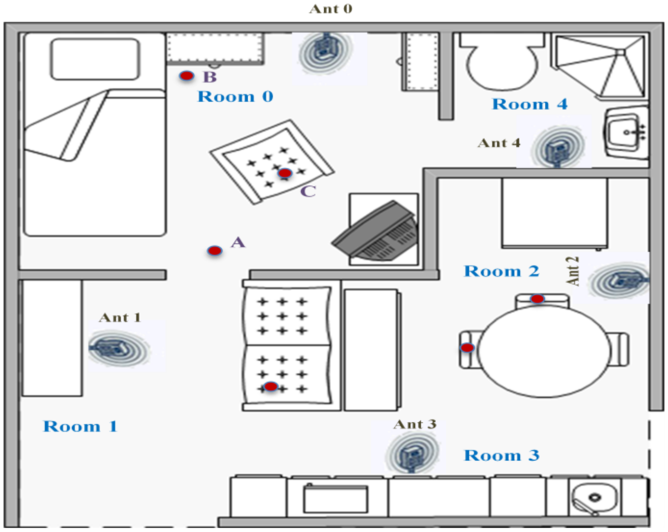
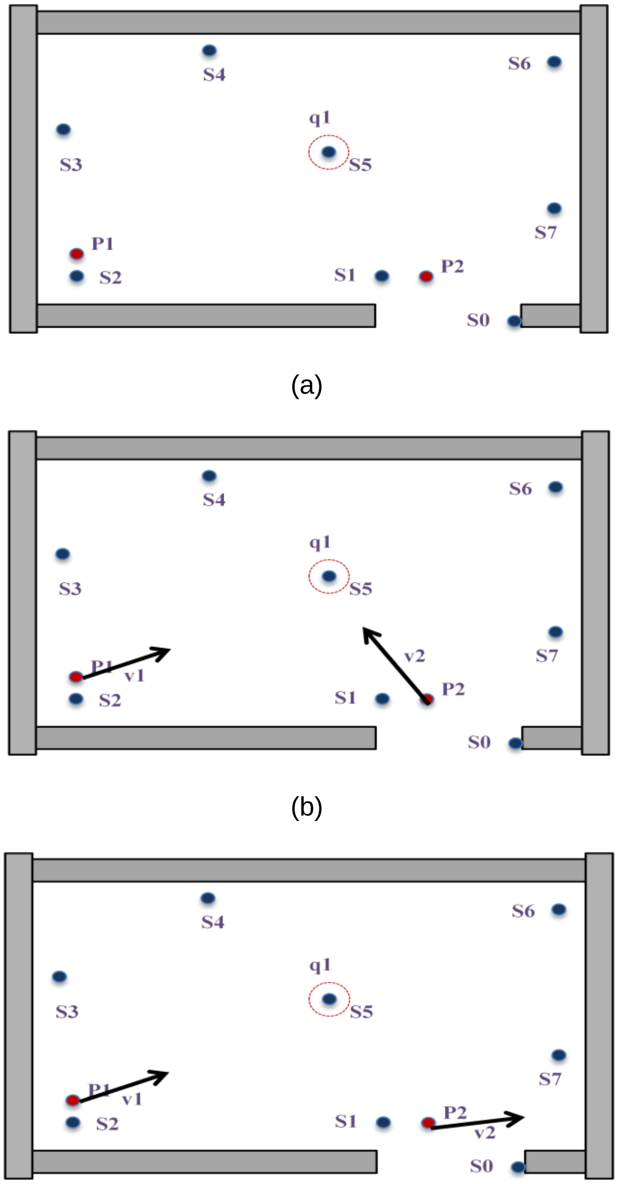


Figure 5.1 Example apartment layout with RFID antennas and sensors

5.2 Identification and Tracking using RFID and Passive Sensor Technology

5.2.1 Identification using RFID Data

As part of the application setting, we deploy one RFID reader antenna per room that covers the entire area of that room. Each person carries a unique RFID tag, which is either attached to his/her ID-card or embedded to the cell phone. Whenever a person enters in a room or more specifically, comes within the range of the antenna deployed in that room, the reader detects a new tag for that person for the first time. If a door-side sensor activates at the same time, the system updates the initial location of that person to the entrance (door-side) of that room. However, if a door-side sensor activates again and some previously detected tags are found to be missing from the reader data for a certain amount of time, the system then assumes the person with those associated tags have left the room.



(a)

(b)

(c)

Figure 5.2 a) An example scenario of 2 persons, P_1 and P_2 at sensor locations S_1 and S_2 at time t and a sensor, S_5 that activates at time (t+1) near point q_1. b) One possible scenario, where P_2 is more probable to activate the sensor S_5 than P_1. C) Another example scenario of P_1 being the most probable persons to activate the sensor S_5

5.2.2 Representing Location Uncertainties

Given the initial location of each person, P_i is known at time t and the maximum velocity is calculated using the history of data, the system represents the location uncertainty of a person at the next time t_n using a circle with radius r equals (t_n−t)×v_{im} [58], where v_{im} is the maximum velocity of each person. The location uncertainty can be even reduced to a half circle, if the approximate direction of movement of a person is known, as shown in Figure 5.3. In our system, we use commercially available Alien 9900+ developer kit [59] that returns the speed and approximate direction of a moving person over time.

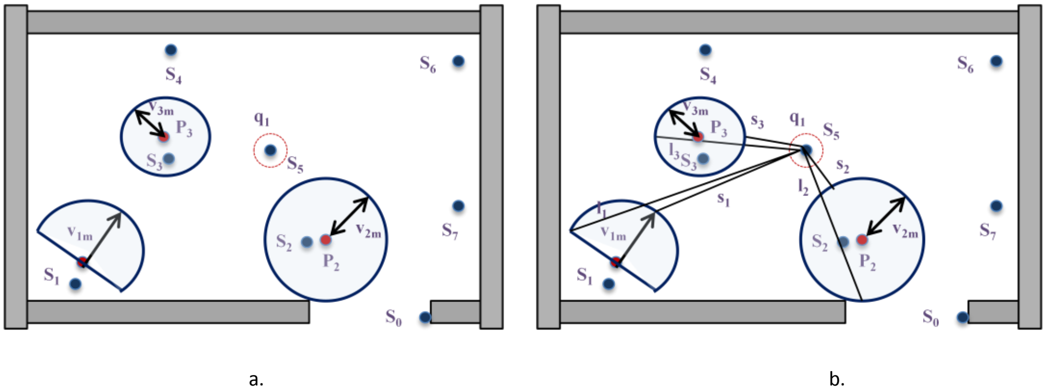


Figure 5.3 (a) Location uncertainty representation of each person at instant t_n and (b) Computation of longest and shortest distances of each person from q_1

5.2.3 Location Update

The location update procedure works as follows:

• Given the uncertainty region of a person and a sensor activates at point q_1, the system computes both the shortest and longest possible distance, denoted as s_i *and* l_i respectively, of each person from the location q_1.

• The system next computes the shortest distance, l_s of all these longest possible distances (l_i's) and draws a circle with radius l_s as shown in Figure 5.4, which defines the probable zone of consideration. Any uncertainty region that falls completely outside of this bounding region is not considered for the location update with regards to the sensor activation at q_1. On the contrary, if the uncertainty region of a person lies completely within the bounding circle, his/her location will be updated with q_1 at time t_n.

• However, if an uncertainty region of a person partially overlaps with the bounding circle, the system then computes the total probability of that person being the nearest neighbor to q_1 for every point of his/her overlapping region with respect to others. The system then compares the probability of each person being the nearest neighbor to q_1 at time t_n and updates the location of a person with q_1, if he has the highest probability of being the nearest to q_1 on that instant.

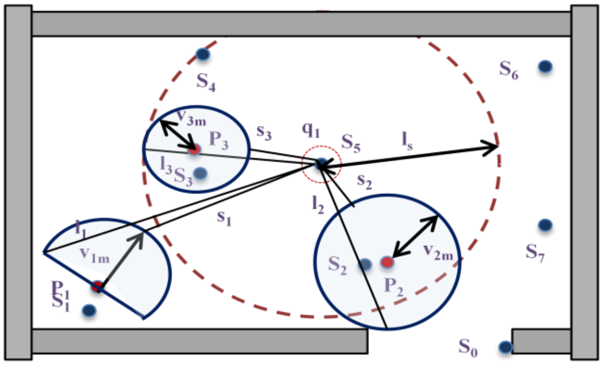


Figure 5.4 Bounding circle representing the probable zone of consideration

5.3 Identification and Tracking using RFID and Kinect Sensor

Table 5.1 Kinect Data Set

User	Time	X Cord.	Y Cord.	Z Cord.
6	4/25/2012 1:22:28 AM	-0.1936212	0.1681233	3.099599
6	4/25/2012 1:22:30 AM	-0.08460984	0.08594385	3.164108
6	4/25/2012 1:22:34 AM	-0.4662972	0.07648824	2.894816
6	4/25/2012 1:22:40 AM	-0.5278196	-0.0273742	3.011885
6	4/25/2012 1:22:45 AM	-0.4450822	-0.1373514	2.829979
6	4/25/2012 1:22:50 AM	-0.456997	0.07926513	2.96067
6	4/25/2012 1:22:53 AM	-0.4790949	0.08204032	2.961271

In this setting, we combine the identification of RFID with the location measurements from the Kinect sensor [56]. Each RFID Antenna has a field of sensing (FOS), within which it can detect a tag. Whenever a person enters the field of view (FOV) of a Kinect sensor and the FOS of a particular antenna, the Kinect and the RFID system capture the location and identification information of that particular person respectively. For a single person, the Kinect sensor itself is sufficient enough, since it can track that person as long as he/she remains in its FOV. The data captured by the Kinect is shown in Table 5.1.

Table 5.2 RFID Data Set

ID	Time	Antenna	RSSI
E2009037890401091080A8BA	4/25/2012 1:22:52 AM	1	4792.4
E2009037890401090900BD64	4/25/2012 1:22:52 AM	2	1351.6
E2009037890401090900BD64	4/25/2012 1:22:53 AM	1	1021.2
E2009037890401091080A8BA	4/25/2012 1:22:54 AM	1	4920
E2009037890401090900BD64	4/25/2012 1:22:54 AM	2	1299.1

Nevertheless, the problem becomes very challenging if the number of people detected from the RFID antennas is more than one. In that case, the Kinect sensor may track people within its FOV, but fails to discern between them. The reason is that the User/Tracking ID from the Kinect sensor is regenerated every time a user leaves and re-enters the scene. As a result, the tracking ID from the Kinect is not unique. On the other hand, RFID antennas capture a unique ID of each person carrying a tag, who is in the FOS of the antenna. Therefore, the main challenge is to map the tracking information of different users captured through the Kinect with the identification information of RFID. However, RFID readers also provide useful information in

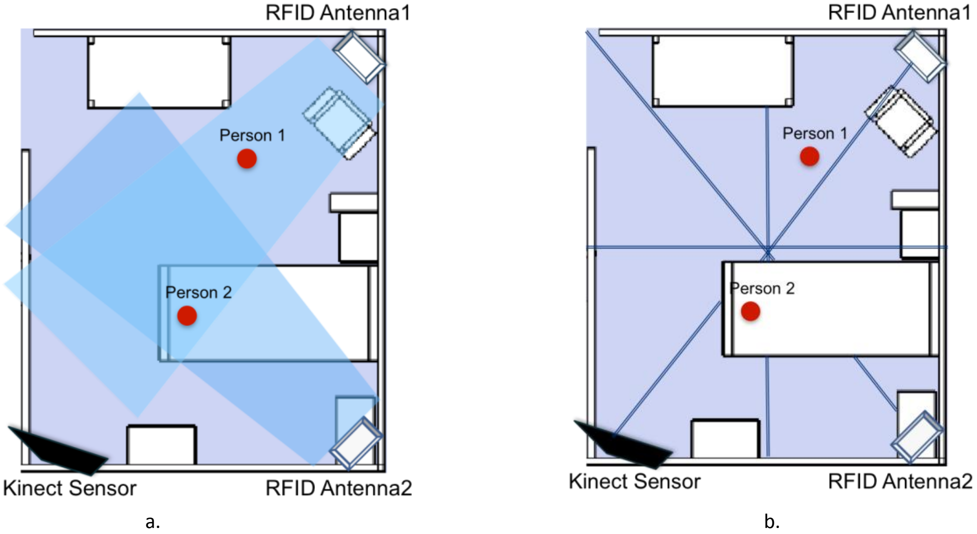


Figure 5.5 (a) Proximity-Based Approach, (b) Classification-Based Approach

the form of the RSSI, which can give an approximate measurement of the distance between the antenna and the RFID tag as shown in Table 5.2.

Therefore, our system utilizes the RSSI so as to increase the accuracy of the localization problem. We have considered two approaches to solve this problem: 1) Proximity-based and 2) Classification based, as shown in Figure 5.5 and described in the following subsections.

5.3.1 Proximity Based Approach

According to this approach (Figure 5.5a), we first record the RSSI signature of the detected tags from each reporting antenna to the locations of interest, such as bed, chair, couch, near bed, between bed and chair in the bedroom area etc. Next, we build a proximity database that describes the signature properties of each of the locations. The properties that we store in the proximity database are as follows:

- $Ant_i^{Loc_j}$, Where $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$ and Loc_j denotes a particular location, such as bed, chair, couch etc. Note that m defines the total number of distinct locations that we will consider for our tracking and n represents the total number of RFID antennas present in the system. The value of the attribute: $Ant_i^{Loc_j}$ can be either true or false, which represents whether a tag can be read by Ant_i, given the location of the tag is Loc_j.

- The next set of attributes is: $\min_{Loc_j} RSSI^{Ant_i}$ and $\max_{Loc_j} RSSI^{Ant_i}$, which represent for each Loc_j, the minimum and maximum RSSI captured from an antenna Ant_i.
- For each location Loc_j, we also record the minimum and maximum X, Y, and Z value recorded through a Kinect sensor, denoted as: $\min_{Loc_j} X^{Kinect_k}$, $\max_{Loc_j} X^{Kinect_k}$, $\min_{Loc_j} Y^{Kinect_k}$, $\max_{Loc_j} Y^{Kinect_k}$, $\min_{Loc_j} Z^{Kinect_k}$, $\max_{Loc_j} Z^{Kinect_k}$. Here, $k = 1, 2, \dots p$, where p represents the total number of kinect sensors present in the system.

For tracking, we consider a window of one second that contains the combined data from both the RFID events stream and the Kinect skeletal tracking stream. From the combined data stream within the current window, we then calculate the total number of distinct users along with their ID, who are present in the scene on that particular second. Next, we consider every data point in the current stream and estimate the location signature for each user. The procedure is as follows:

For each data point in a specific time window, we first check whether the data point is an RFID event or a Kinect tracking event. If it is an RFID event, we record the RFID Antenna that detects the particular user. For example, if a user X is detected through Antenna 1, the system will record, $X_{Ant_1} = true$. Now, if the user is detected for the first time in the current window, the system records the observed RSSI ($X_{Ant_i}^{RSSI}$) as well. But, if the user has been detected before, the system first checks whether it is the same RFID antenna that has detected the tag. If yes, it simply updates the user attribute info with the maximum observed RSSI between the current and previous record. For example, if the same user X is detected by antenna 1 again, the system compares:

$If\ X_{Ant_i}^{RSSI_{current}} > X_{Ant_i}^{RSSI_{prev}}$,

$$then\ X^{RSSI}_{Ant_i} = X^{RSSI_{current}}_{Ant_i}$$

However, if the current antenna is different than the previously recorded one, the system checks the true flag for the new antenna and records the corresponding RSSI for that particular user at that particular time. Now, if the data point comes from the Kinect, the system first records the X, Y and Z coordinates and then, for each user found in the system, searches for a location in the proximity database, for which the signature matches the most to the observed pattern. Then the location of that particular user is updated with the location recorded through the Kinect in the form of X, Y and Z values. The pseudo code of this approach is given in Algorithm 1.

Algorithm 1 Proximity Based approach

```
Build the Location Proximity Database:$Ant_i^{Loc_j}$,
$\min_{Loc_j} RSSI^{Ant_i}$,$\max_{Loc_j} RSSI^{Ant_i}$,$\min_{Loc_j} X^{Kinect_k}$,$\max_{Loc_j} X^{Kinect_k}$,$\min_{Loc_j} Y^{Kinect_k}$,
$\max_{Loc_j} Y^{Kinect_k}$, $\min_{Loc_j} Z^{Kinect_k}$, $\max_{Loc_j} Z^{Kinect_k}$
Combine data within one second in the current window
Calculate the total number of distinct users along with the IDs
**for each** data point in the current window **do**
**if** the data point is an RFID event **then**
    Note the Antenna, $i$ that detects the tag
    **if** no previous event exists for that user **then**
            $X_{Ant_i} = true$
            $X^{RSSI}_{Ant_i} = RSSI_{current}$
    **else**
        **if** $Ant_i\ equals\ Ant_{previous}$ **then**
            **if**$X^{RSSI_{current}}_{Ant_i} > X^{RSSI_{prev}}_{Ant_i}$**then**
                $X^{RSSI}_{Ant_i} = X^{RSSI_{current}}_{Ant_i}$
            end if
        else
            $X_{Ant_i} = true$
            $X_{Ant_i}^{RSSI} = RSSI_{current}$
        end if
    end if
else
    Record the X, Y and Z values of the Kinect
    For each detected user do
        for each location record in the Proximity Database do
            Find out the location, for which the attribute matches the most with the
            signature of the user and the recorded location values X, Y, Z.
            Record that particular user and exit the inner loop.
        end for
        Update the location of that particular user with the X, Y and Z coordinates
        recorded through the Kinect and exit the outer loop.
    end for
end if
end for
```

5.3.2 Classification Based Approach

In this approach, we divide the entire apartment or individual room into multiple sectors, as shown in Figure 5.5b. Next, we collect the RSSI signatures of the detected tags in these different sectors using the antennas. In the training phase, we label the signatures with their corresponding sector number and apply a statistical regression method (a fitted model to describe the relationship between the selected and the observed values as well as to predict newer values) to build a classifier that classifies any RSSI measurement from an antenna into one of these different sectors. The idea is that given an RFID event is detected, the system first classifies that event to narrow down the location of the detected tag into any one of these sectors. Next, given the measurement from the Kinect sensor for any particular user, if the measured location falls within that specific sector, then we map that particular user who generated the RFID event to the location described by the Kinect sensor. A shortcoming of this approach is that if the two users remain in the same sector meaning, they are in close proximity, the exact identification with accurate localization may fail. Nevertheless, the system will detect the approximate locations, such as sector 1 or 2 for both users correctly.

5.3.3 Improvements for both Approaches

In both of the above-mentioned approaches, we map the RFID event to a Kinect event in every second. Realistically, a Kinect sensor can recognize up to 6 persons simultaneously. Once a person is mapped to a particular location using RFID signatures and Kinect tracking information, that person can be simply tracked using the Kinect as long as the person remains in the FOV of that Kinect. However, if the person leaves the FOV and re-enters, a new tracking ID is assigned, which needs to be re-mapped. But, the problem becomes simpler, if only one person leaves the scene and later re-enters (all others remain in the FOV of the Kinect). In that case, if the RFID antenna still detects that person’s tag and the Kinect detects a new skeleton with a newly assigned ID, we can simply map that skeleton to the RFID tag. But, if more than one person enters the scene and two or more unassigned RFID tags are detected, we can apply either proximity or classification based approach to find out the correct mapping.

5.4 Result

In our experimental setup, we have used two RFID antennas and one Kinect sensor in the bedroom area of our simulated assistive apartment at the Heracleia lab.

5.4.1 Accuracy

We have run our system in the simulated bedroom of the Heracleia Assistive Apartment to identify and localize up to four people simultaneously. For each location update from the

Kinect, our visualization component transforms the world coordinates into screen coordinates to display the location of each person correctly in the 2D apartment map shown in Figure 5.6.

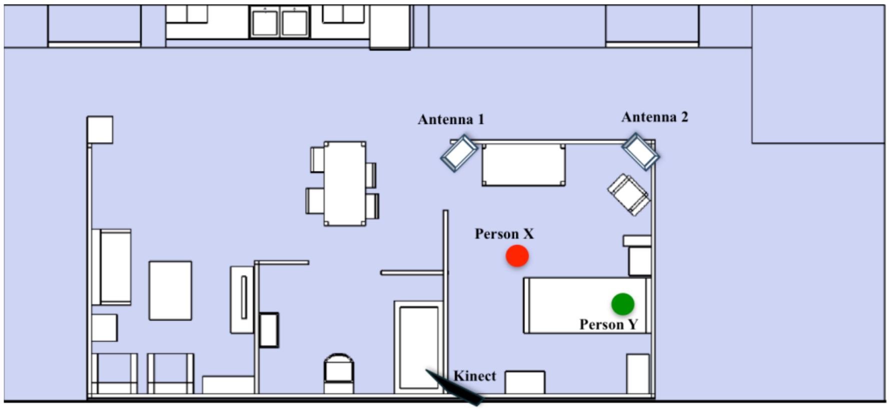


Figure 5.6 2D layout of our simulated apartment with the RFID Antennas and Kinect setup

The accuracy of our system for the two approaches considered in this paper is shown in Table 5.3. Note that, for the classification-based approach, we have divided the entire room in 8 sectors (shown in Figure 5.5b) and applied 10-fold cross validation to calculate the accuracy. Here, the accuracy specifies the percentage of correctly predicted locations for each person present in the room (the entire room is divided into 8 sectors). As shown in the table, the proximity based approach gives higher accuracy than the classification based approach with/without improvements. The reason is that in proximity based approach, we compare each Kinect event against the unique RFID tags and signal strengths observed on that particular second, as well as against the ranges of signatures observed in a particular location (sector). In the classification-based approach, we only consider the RFID signal strength for deriving the sector and then, we provide the mapping.

Table 5.3 Experimental Results

Approach	Accuracy
Classification based on Statistical Regression (4-person)	60%
Classification with improvements (4-person)	65%
Classification-based (2-person)	67%
Proximity-based (4-person)	68%
Proximity-based with improvements (4-person)	76%
Proximity-based (2-person)	86%

5.5 Discussions

In this chapter, we first proposed a fundamental technique to identify and localize multiple person in a less-intrusive manner using existing RFID identification and passive sensor technology. We also combined the localization capabilities of the Kinect sensor with identification information from existing RFID technology. We utilized the high-level location information and RSSI property to resolve the localization and identification of multiple people simultaneously. We considered two methods: 1) proximity based approach and 2) classification based approach as a solution to this problem and ran experiments in a simulated room of an apartment. Our experiments proved the effectiveness of our system for the 4-person localization scenario. Future extensions will include the integration of multiple Kinect sensors as well as the utilization of the Kinect's microphone array.

CHAPTER 6

AUTOMATED MONITORING OF MEDICINE INTAKE

6.1 Introduction

Monitoring medication intake of a person, who needs assistance has become an important research arena recently. Statistical report has shown that almost 55 percent of the elderly people in US fail to adhere to their daily medication routine [60]. Among these, 26 percent of the errors are severe. As patients forget to comply with their prescription, they need constant care from doctors, nurses or other caregivers. Therefore, at home pervasive and assistive monitoring systems with minimal intrusion into a person's personal life can be very effective in this scenario. The *Smart Drawer* is one such application that helps patients in maintaining their medication intake as consistent as it is prescribed by a healthcare professional to them. Besides reminding a patient to take his or her pills on time, the system also logs all the activities of the patient for further analysis in case he or she fails to obey the prescription [52]. The Smart Drawer system includes: 1) sensors to detect the open or close state of the drawer, 2) sensors to detect when a medicine bottle has been removed/placed back in the drawer and 3) a precision balance to detect how many pills has actually been taken out of the bottle after removing it from the drawer. In the context of a pervasive environment, such an application with multi sensor data fusion improves the inferring capability of the system more than it could achieve using single sensor alone [61]. Perhaps, the most appropriate example of multi-sensor data fusion would be the one that is naturally performed by humans and animals. A human uses the combination of touch, smell, taste, vision and hearing capability so that they can access their surrounding environment better to improve the overall chances of survival.

In this chapter, we describe the Smart Drawer system that records the removal of each individual medicine bottle from the drawer as well as the time when the medicine is actually taken. By doing so, the system can track the most common problems in patients and helps to treat them better according to their medical history. The system also tracks prior and subsequent activities of a person before they remove a pill from the drawer to derive useful information about the person's behavior. For example, a person may open or close the drawer several times, but forgets to take the medicine from it. On the other hand, the person may remove the medicine out from the drawer, but put it back again without consuming a pill. Such information could give very good indication about a person's behavior.

6.2 Approach

In this section, we explain our approach by describing the system architecture, the layout of the experimental setup, a state diagram followed by a sequence diagram.

6.2.1 System Architecture of the Web Tool

The Smart Drawer project implements a web based tool for an RFID reader system, where it first establishes a connection by imposing some constraints on the hardware. However, the web-based tool offers three different views: 1) a caregiver view, 2) a maintenance view and 3) a patient view. Here, the caregiver view displays the details about the history of drug taking patterns of the patients. The maintenance view reports the working status of different sensor nodes over time. The patient's view includes an alert system that prompts the patients to take their medication. Now in an assistive living environment, a patient is supposed to take different types of pills each day. Therefore, detecting the sequence of medicine intake by creating a pattern helps the patients to take their medicine on time. Such a system also helps the patients in taking the right number of pills, by incorporating the precision balance. On the other hand, once the caregiver/doctor log on to the system by authenticating themselves as the legitimate users of the web based Smart Drawer tool, they can see all the details of their patients and verify, if the patients have taken the right amount of medication at the right time. They can also add new patients and their details as well as compare the medication details with the backend database. In short, the Web Tool Architecture shown in Figure 6.1 has two parts: 1) the web module and 2) controller module. The Web module consists of the pages containing caregivers/doctors login and display of patient's medication intake records as well as patterns. The Controller Module acquires the heterogeneous data from all the sensors operating in the Wireless Sensor Network Environment and determines the Pattern using an algorithm like Dynamic Time Wrapping algorithm [62]. The Controller Module also compares the pattern with a dictionary of normal behaviors to recognize whether the behavior is normal or abnormal.

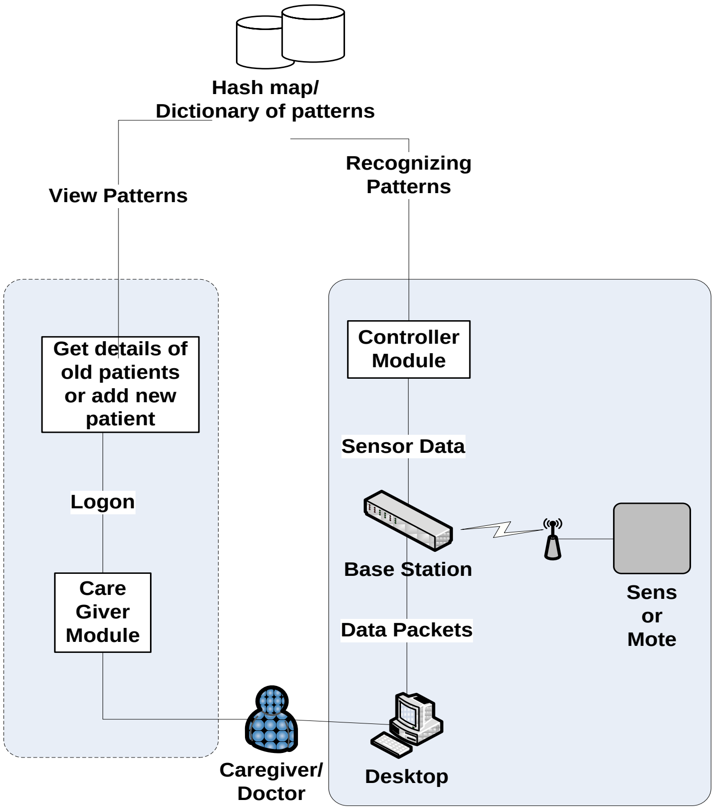


Figure 6.1 Web Tool Architecture

6.2.2 Architecture of Data Fusion Model

The Data Fusion model as shown in Figure 6.2 contains: 1) a Human computer interface module, 2) a module containing all sensors, 3) a source pre-processing module and 4) database management module [61]. However, the Database Management System consists of two components named Support Database module and Fusion Database module. For the purpose of Data Fusion, we consider information from only the RFID reader, SunSPOT [63] device and a precision balance. Here, the RFID reader generates a continuous stream of tuples identifying the presence of a medicine bottle inside the drawer. The SunSPOT device generates data to represent the drawer opening/closing status at a particular moment. Finally, the precision balance generates data representing the total weights of medicine bottles in the drawer at a specific time.

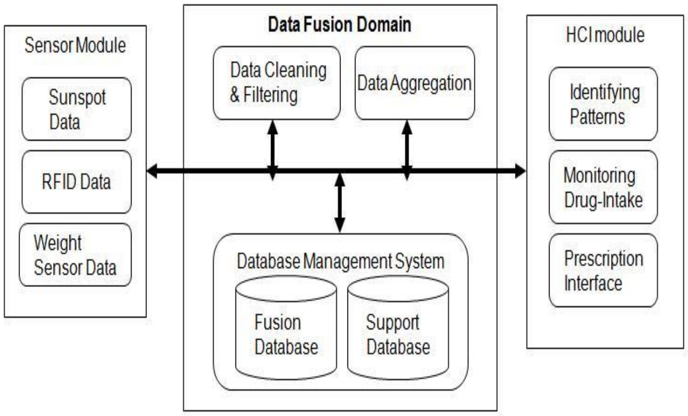


Figure 6.2 Data Fusion Architecture

The HCI (Human-Computer Interface) module contains high-level user interfaces for inserting prescriptions, monitoring drug intake and gathering statistics. We believe it is the responsibility of the caregivers, doctors or the families that they provide accurate prescription information to the system. But, the system at least provides a flexible interface to them for adding these prescription data with minimum effort. The HCI module also generates alerts or reminds caregivers and patients in case a patient deviates from their prescribed medication pattern. Moreover, the system summarizes all the activities of a patient at different time via HCI so that doctors and caregivers can analyze that information to identify useful patterns. The Data Cleaning and Filtering module is a very important component as sensors generate a large volume of data; most of which may be irrelevant to the current context. Moreover, data generated from sensors contain both redundancy and error. Therefore, appropriate filtering techniques must be adopted in order to remove this redundant information. Reliable data cleaning techniques should also be applied to the data in order to correct any errors. The Data Aggregation module combines all the preprocessed data from different sources in the form of (drawer open time, drawer close time, RFID event start time, RFID event end time, RFID tag, weight changes). However, the Support Database component of the Data Fusion module contains information about doctor's prescription for a patient, patient's current medical condition as well as his or her personal information. It contains medication records, total number of bottles per medicine, total weight per bottle along with its associated RFID tag and expiry date as well. Finally, the Fusion Database contains all the aggregated information generated from the continuous interaction of patients with the sensors. The Database Management System also provides support functions to access the fused data as well as to make queries about it.

6.2.3 State Diagram and Sequence Diagram

6.2.3.1 Sensor Combining State Diagram

Figure 6.3 shows our drawer system consisting four states, which correspond to the different types of sensors involved in the testbed. S1 represents the start state, when no motion is detected within the room containing the medicine drawer. Once the motion detector is triggered by the presence of a user, the machine makes transition to state S2. However, state S2 remains unchanged until it is triggered by a drawer event (open/close), where it changes the state to S3. State S4 indicates that a medicine bottle has been removed from the drawer. As soon as the medicine bottle is placed back in the drawer, the balance records the new weight.

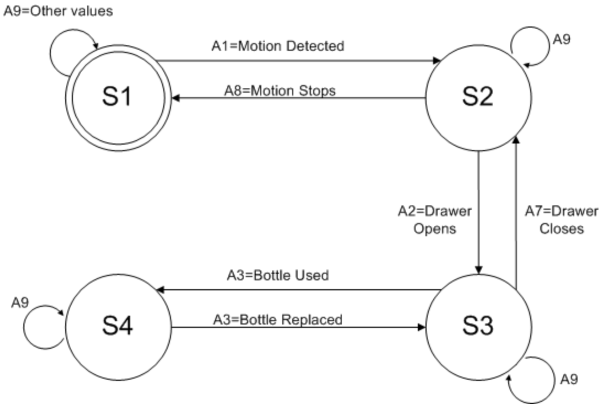


Figure 6.3 Sensor State Machine

6.2.3.2 Sequence Diagram

As shown in the sequence diagram of Figure 6.4, the system starts scanning for the available RFID tags in the drawer as soon as a patient opens it. Now, if the reader misses a tag in the drawer for a while, the system scans for this missing RFID tag until it detects it again. The system then creates a corresponding RFID event from the raw RFID data that represents the amount of time a medicine bottle was missing from the drawer. The system also measures the weight difference of the medicines at the end of that interval. However, if the weight changes within this time, the system records that difference and logs the time, which is eventually displayed to the caregivers through the monitoring medicine intake interface. The system also keeps track of all the activities, such as individual drawer event, RFID event in order to identify useful information about the patient's behavior. The system finally summarizes the entire drug taking activities of a patient by comparing whether it is close to normal or deviating from the normal and displays such information to the caregivers or doctors through the "generate statistics" interface.

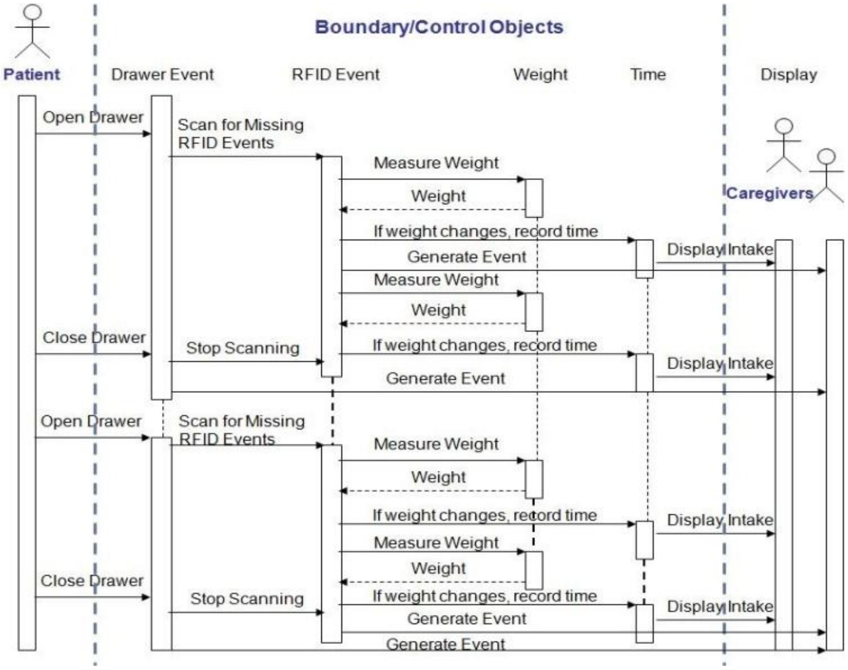


Figure 6.4 Sequence Diagram

6.3 Results and Snapshots

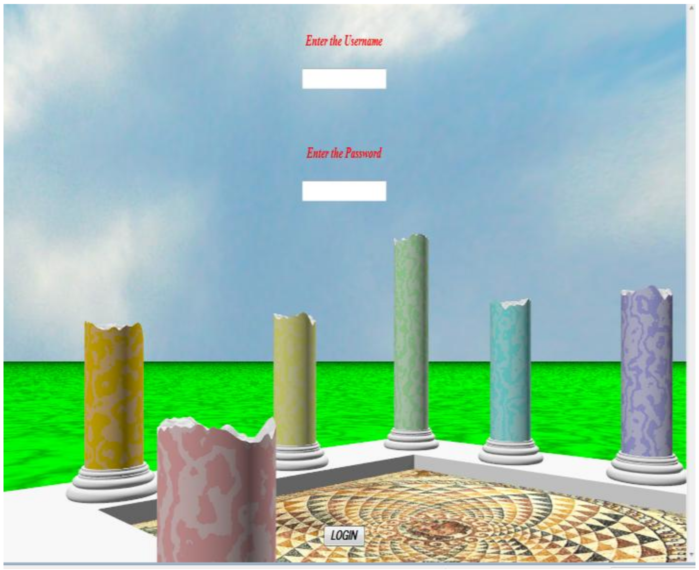
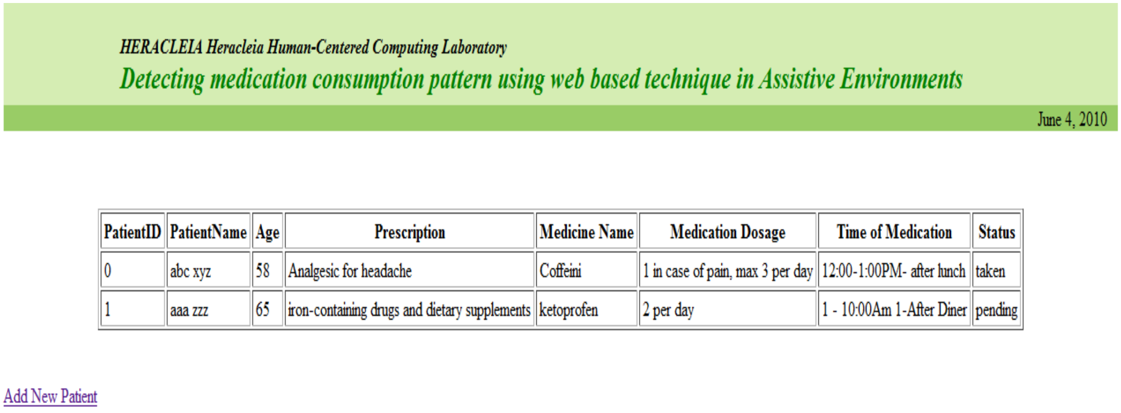


Figure 6.5 Login Page

The system is designed such that, on a successful login to the page as shown in Figure 6.5, the caregiver can view the details of the patients, as described in Figure 6.6.



HERACLEIA Heracleia Human-Centered Computing Laboratory

Detecting medication consumption pattern using web based technique in Assistive Environments

June 4, 2010

PatientID	PatientName	Age	Prescription	Medicine Name	Medication Dosage	Time of Medication	Status
0	abc xyz	58	Analgesic for headache	Coffeini	1 in case of pain, max 3 per day	12:00-1:00PM- after lunch	taken
1	aaa zzz	65	iron-containing drugs and dietary supplements	ketoprofen	2 per day	1 - 10:00Am 1-After Diner	pending

Add New Patient

Figure 6.6 Patient details viewed by the Caregivers

Such details include: 1) the time at which the patient has taken the medication, 2) the name, type and amount of medication taken. However, the system measures the amount of medicine taken in a particular round, by comparing the weight difference (shown in analytic balance sensor) between all the medicine bottles, after a bottle is removed from the drawer and placed back again. Note that, the analytic balance used in the system is able to detect changes in weight to the order of milligrams.

6.4 Discussions

The major contribution of the Smart Drawer system includes the web-based medicine tracking tool. The tool plays a crucial role from the care giver logins being the primary authentication step till the determination of the quantity or the amount of medication consumed by the patient at the prescribed time. This makes the system accessible from a remote workplace, where the caregivers can track the medication using the web interface. The tool also helps the caregivers to identify the behaviors of the patients according to their medicine intake patterns.

CHAPTER 7

AUTOMATED ASSESSMENT OF DEPRESSION FROM TEXT

7.1 Introduction

Major Depression is a severe mood disorder that affects both physical and cognitive functioning of a person (patient) while limiting his/her physical and emotional recovery [64]. Such a disorder is very common in the adult population and the severity level may depend on a patient's prior stroke records or brain injury [65]. However, the task of automatic assessment and monitoring of depression symptoms is still dependant on the paper-and-pencil based homework (HW) assignments, where a person is asked to write his thought records (TR) in plain text and rate his current emotional state. A TR is a loosely structured document consisting of emotional expressions and feelings of a patient with respect to any recent stressful events, which is later examined by the clinicians or therapists manually. But, the process of manual examination by the therapists is very expensive with regards to both cost and time. Nevertheless, the accuracy of the clinician's rating about a patient's depressive symptoms also depends on his/her overall skill and experience level. Therefore, our system includes an automatic depression assessment tool that uses collection of text-based homework records such as TRs. Given, the thought records filled out by different individuals, our system automatically classifies the text records into one of the two main categories: Depressed or Not Depressed. Such step can be utilized as an initial start-point for a therapist to work further.

To train our system with sample data points for non-depression, we use the corpus of natural conversations collected in text format by the University of Santa Barbara. For the depression records, we use the anonymous thought records of different patients from the Psychology department of UTA. We consider each term or word appearing in a text document as a feature and apply different feature selection methods such as information gain to select the best feature set. We later apply different classification schemes such as SVMs to classify the text records into one of the two categories. Our system obtains approximately 90% classification accuracy [66] for the dataset containing 77 sample points for depression documents and 55 sample points for natural conversation.

7.1.1 Background

According to American Psychiatric Association 2010, Cognitive Behavior Therapy (CBT) is considered to be an effective first line treatment of depression for the patients, who are non-adherent to antidepressant medication. Homework (HW) in CBT, which is a list of activity assignments that a patient performs outside the office, has a positive effect on the therapy session. Such an assignment helps the patient to exercise what is learnt during the therapy session and also develop and improve skills to identify negative thoughts through practice. The hallmark CBT homework assignment is a thought record (TR). TR is a critical tool for the therapy process that allows patients to express negative automatic thoughts and emotional reactions while describing their recent stressful life events. The information in TR helps the therapists to understand and select the interventions to be used in a therapy session that would reduce a patient’s depressive symptoms. But, such a process is very challenging, as the tasks seem less enjoyable to the patients and thus, remains incomplete most of the time.

Various attempts to automate the process of depression assessment have been presented in the literature. Most approaches try to determine the emotional state of the subject using different types of information. Among them, facial features are the most popular while vocal features yield significant results as well. Furthermore, advanced techniques use both types of features in a multimodal implementation in order to increase robustness and accuracy [67, 68, 69, 70]. Text is also informative concerning the emotional state of the subject by determining the use of specific words or grammar rules. There are variety of methods for analyzing text content such as keyword and phrase spotting [71], rule-based modeling [72], Semantic Trees [73] and N-Grams [74]. Nevertheless, depression assessment is a higher level process that can be more challenging than emotion recognition. So far, most researchers have used other types of data as indicators for depression. Magnetic resonance imaging data in conjunction with machine learning techniques such as Support Vector Machines have been used for diagnosis and prognosis of depression too [75]. Alternatively, SVMs were used with functional MRI data used to monitor neural responses to verbal fluency in order to detect depression and other disorders [76]. An alternative approach uses facial expression and vocal prosody with SVMs on videos from a clinical trial to detect depression [77]. In our work, we follow an alternative approach by focusing on information extracted from text in order to detect depression.

7.2 Methods

7.2.1 Data Collection and Preprocessing

7.2.1.1 Collection of Thought Records

We have collected anonymous homework records, known as TRs completed by the patients outside therapy session, from our psychology department at UTA. The dataset contains plain text written by the patients, which describe their current mood or feelings and automatic negative thoughts, or experiences from any recent events of frustration.

7.2.1.2 Corpus of Natural Conversation

As a reference dataset for non-depressed dialogue, we have used the corpus of natural conversation from the University of Santa Barbara that contains hundreds of recordings of natural speech from all over the United States. Each speech record also contains its associated transcript file, where dialogues are time-stamped and ordered according to the audio recording. The names or identity information used in all these transcripts are fictitious. Also, the audio portions that contain identity information are filtered to be unrecognizable to preserve the anonymity of the speakers. The dataset contains a wide variety of conversations among people of different regional origins, ages, occupations, and social backgrounds. Such conversations include natural talk, arguments, on-the-job talk, games, meetings, sales pitches, lectures, stories, etc.

7.2.2 Feature Selection

7.2.2.1 Speaker Separation

Each audio record as well as the transcript file from the corpus of natural conversation contains speech records of multiple speakers. Therefore, given the records of mixed conversation, we first parse the speech from different speakers into separate files.

7.2.2.2 Stop Words Removal and Vocabulary Construction

We have identified approximately 150 stop (common) words, such as “the”, ”so”, “an”, “he” etc, which do not convey any meaningful information in detecting depression. Therefore, from the raw text record of each individual speaker, we identify and remove the stop words. In addition, we also remove the words that appear for less than 3 times in the overall record set. Finally, we add the rest of the words to a unique vocabulary set, called dictionary. Such dictionary contains all the unique words or terms that have appeared in the raw text record either as a normal conversation or depression thought records.

7.2.2.3 Term Frequency and Inverse Document Frequency (TF-IDF) Computation

We consider each word or term present in the dictionary as a feature for our classification and compute their corresponding TF-IDF weights. TF-IDF is a well-known information retrieval technique, where the term TF represents the occurrence of a word, w or the frequency count of a feature in a document, d ($n_{w,d}$) with respect to the total number of words (n_d) present in it. Therefore,

$$TF_{w,d} = \frac{n_{w,d}}{n_d}$$

The TF scheme considers each word or feature to be equally important, which is less effective in many cases as certain words or features convey more information than others. To solve this problem, the Inverse Document Frequency or IDF of a word is computed that reflects the relative importance of a word with respect to some documents. In short, IDF scheme provides higher importance to the words that appear in less number of documents. According to the definition, we first compute the DF (document frequency) for each word in the dictionary, denoted as DF_w. This means, for each word, we count the number of documents (DF_W) that contain the word, w. Next, given the total number of documents in the collection is N, we compute the inverse document frequency of a word (IDF_w) using the following equation:

$$IDF_w = \log \frac{N}{DF_w}$$

Finally, we compute the TF-IDF weight of each word in a document as

$$TF - IDF(w_d) = TF_{w,d} \times IDF_w$$

7.2.3 Classification

7.2.3.1 Cross Validation

We apply a well-known statistical technique, called 10-fold cross validation to measure how accurately our classification system works. According to the 10-fold cross validation approach, for each fold/iteration, the entire dataset is partitioned into 10 equal sub-groups. Among these 10 sub-groups, 9 are selected for the training purpose and the rest one is used for validation or the testing purpose. Thus, in each round, the system takes disjoint training and test data set. However, the same operation is repeated for 10 times and the final classification accuracy is measured by taking the average of the accuracy from each fold. Since multiple rounds of partitioning and classification are considered, the approach reduces variability of the final classification result.

7.2.3.2 Feature Selection Methods

7.2.3.2.1 Information gain.

Information Gain (IG) measures the number of bits of information obtained for class prediction by knowing the value of a feature. Let $\{c_i\}_{i=1}^{m}$ denote the set of classes. Let *V* be the set of possible values for feature *f*. The information gain of a feature f is defined to be:

$$G(f) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + \sum_{\upsilon\in V}\sum_{i=1}^{m} P(f=\upsilon)P(c_i \mid f=\upsilon)\log P(c_i \mid f=\upsilon)$$

7.2.3.2.2 Chi-Squared.

The *χ^2-statistic (χ^2)* [78] measures the lack of independence between f and c. It is defined as follows:

$$\chi^2(f) = \sum_{\upsilon\in V}\sum_{i=1}^{m} \frac{(A_i(f=\upsilon) - \mathrm{E}_\iota(f=\upsilon))^2}{\mathrm{E}_\iota(f=\upsilon)}$$

where *V* is the set of possible values for feature *f*, *$A_i(f = \upsilon)$* is the number of instances in class c_i with *$f = \upsilon$, $E_i(f = \upsilon)$* is the expected value of *$A_i(f = \upsilon)$. $E_i(f = \upsilon)$* is computed with *$E_i(f = \upsilon)$ = $P(f = \upsilon)P(c_i)N$*, where *N* is the total number of instances.

7.2.3.2.3 ℓ$_{2,1}$-Norms minimization.

ℓ$_{2,1}$-Norms Minimization, is a feature selection method which reduces the feature dimensionality by performing sparsity regularization on the initial feature set which gives a high weight to the most discriminative features and small weight to the rest of them. The optimal weights (coefficients) are obtained by performing ℓ$_{2,1}$-Norms Minimization on the linear regression objective function. The minimization problem to be solved is:

$$\min_W \; {}^1\!/_{\gamma} \left\| X^T W - Y \right\|_{2,1} + \left\| W \right\|_{2,1}$$

where $X = [x_1, x_2, ..., x_n] \in R^{d\times n}$ is the data matrix, $Y = [y_1, y_2, ..., y_n] \in R^n$ is the vector of labels (classes) and $W \in R^{d\times c}$ is the matrix of coefficients to be computed. For more details on how to efficiently solve this optimization problem, please refer to [79].

7.2.3.2.4 SVM-RFE.

SVM Recursive Feature Elimination (SVM-RFE) is a feature selection algorithm, which selects the top features by deciding a ranking of the features of a classification problem by training a SVM with a linear kernel and removing the feature with smallest ranking criterion at each round. This criterion is the w value of the decision hyper plane given by the SVM. For a more detailed explanation of the algorithm, please refer to [80].

7.2.3.2.5 SVM classification techniques.

Support Vector Machine (SVM) is a standard binary classification technique [81, 82] that selects the best hyper-plane, which provides the largest separation or margin between two classes. Since larger margin between two classes means lower classification error, SVM selects a hyper plane that has the largest distances from any training data of the two classes. Given a set of labeled training data set $(y_1, X_1), (y_2, X_2) \ldots (y_l, X_l)$, where $y_i \epsilon \{-1, 1\}$ and X_i's are input vectors, SVM obtains an optimal hyper-plane that linearly separates the training data set, only if there exists a vector W and a scalar b such that

$y_i(W \cdot X_i + b) \geq 1, i \, \epsilon \, \{1, 2, \ldots, l\} \, (1)$

holds. However, the optimal hyper plane can be derived using

$W_0 \cdot X + b_0 = 0 \quad (2),$

where (W_0, b_0) are the arguments that maximize the margin between the two classes. Now, for the case, where the training data sets are not linearly separable, SVM separates the training set with minimal number of errors, while introducing some non-negative slack variables $\xi_i, \ i = 1, 2, \ldots, l$. In such a case, SVM classifies the training data set based on the equation:

$y_i(W \cdot X_i + b) \geq 1 - \xi_i, \ i \, \epsilon \, \{1, 2, \ldots, l\} \, (3).$

In other words, the goal of the SVM is to minimize the function

$$\frac{1}{2}W^2 + CF(\sum_{i=1}^{l} \xi_i{}^{\sigma}) \ (4)$$

where C is the error penalty, $F(x)$ is the decision function and for a small value of $\sigma > 0$, $\sum_{i=1}^{l} \xi_i{}^{\sigma}$ defines the number of training errors.

Hence, the goal of SVM is to determine the hyper plane with arguments W_0 and b_0 that minimizes the error function as described by (4), as well as provides maximum separation among the rest of the training data from the two classes.

7.3 Experimental Results

We have experimented with 4 different feature selection methods (IG, χ^2-statistic, $l_{2,1}$-Norm minimization and SVM-RFE) to reduce the dimensionality of the feature vector and have used SVM with RBF kernel to perform classification. To find the number of features that gives the highest accuracy, we have trained our classifier with an incremental number of features starting from 100 and going up to 2100 features in increments of 200 every time. In our experiments, $l_{2,1}$-Norm minimization and SVM-RFE have performed comparably and significantly better than IG, χ^2-statistic. The highest accuracy of **92.6%** has been achieved by $l_{2,1}$-Norm minimization with 900 features, although the accuracy was marginally above 90% even with 300 features, which indicates that only a few key words are enough to detect signs of depression in an individual. The results of applying different methods are shown in Figure 7.1.

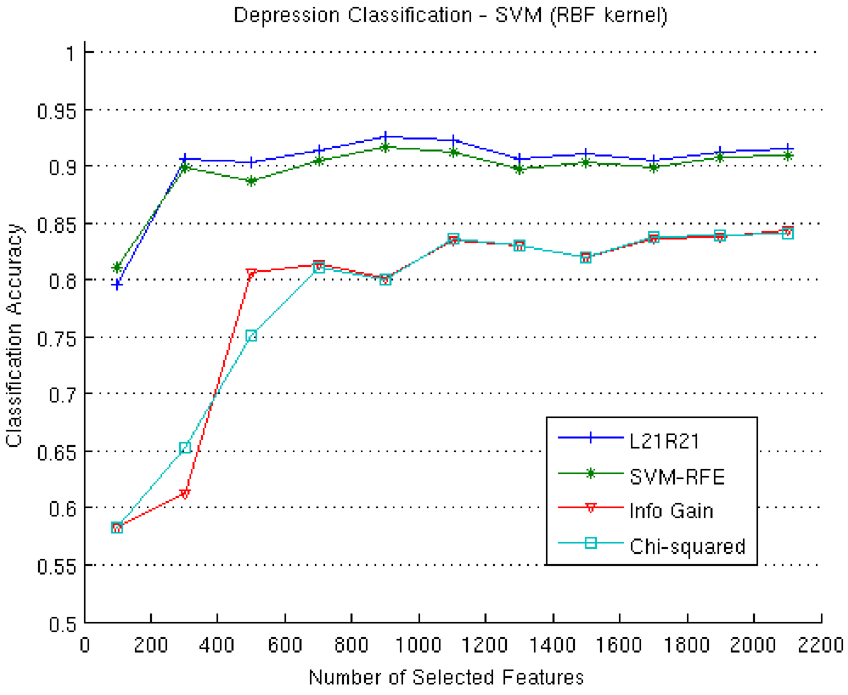


Figure 7.1 Classification Results for Different Approaches

7.4 Discussions

To classify different users as depressed or non-depressed using their text input, we have experimented with a variety of feature selection and text classification methods. Our results have shown that the task of automatically identifying depressed users can be successfully achieved with high accuracy. It means, the way people express themselves in writing can carry important information about their mental state and such information can be utilized by computer programs to identify problematic cases. These findings encourage us to further experiment with computer-aided methods of depression diagnosis and treatment that may facilitate the work of psychologists.

CHAPTER 8

TAGGING AND WAREHOUSING TO QUERY HISTORICAL DATA

Data generated from an assistive application can be of large volume. Now, if a user wants to ask queries about the patient/drug history, traditional database system might not be fast enough to provide fast query responses from these larger volumes of dynamically changing, time-sensitive and location-specific data [50]. Therefore, we need special data structures such as a Data Warehouse built on top of the relational data to provide fast responses over the historical data [44, 85].

8.1 Construction of Data Warehouse

Data Warehousing is a collection of decision support techniques such as OLAP, DSS and Data Mining that enables a knowledge worker/analyst to make better and faster decisions after performing time-series and trend analysis over massive amount of historical Data [83]. Figure 8.1 describes the architecture and subsequent steps necessary to construct a traditional Data Warehouse. In comparison to Traditional Relational Databases, which are optimized to process queries and transactions over fast changing Data Sets, Data Warehouses supports efficient query extraction, processing and analysis for making decision by constructing a multi-dimensional model of storage of integrated Data from multiple heterogeneous sources.

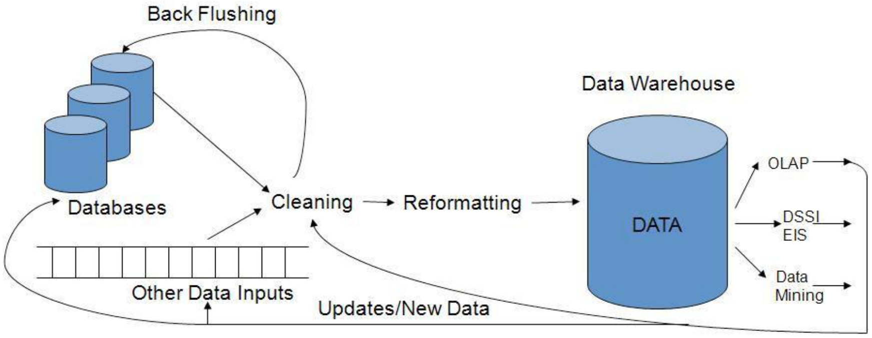


Figure 8.1 Overall Process of Data Warehouse [83]

In the context of our assistive environment, we focus on time series and trend analysis of historical spatio-temporal Data generated from RFID Applications. Therefore, the first level input to our Warehousing System consists of raw RFID Data. But, Data generated from RFID applications contain errors and a lot of redundancy. For example, a single tag can be read multiple times by the same reader in a same location within a very short interval. Furthermore, multiple readers can detect the same tag at the same time in case the distance between these two readers are close and the tag comes close to the vicinity of both readers in the course of its movement. In order to remove the duplicate data without losing any useful or significant information for our analysis, efficient Data cleaning is required over these raw RFID Data.

Table 8.1 Example Raw RFID Data for both Patients and Drugs tracking

RFID Tag	Time	Location
Patient1	T1	L1
Patient2	T1	L1
Patient1	T8	L1
Patient2	T10	L1
Patient1	T11	L1
Patient2	T11	L2
Patient3	T12	L2
Patient2	T13	L2
Patient3	T15	L2
Patient1	T15	L3
Patient1	T20	L3

RFID Tag	Time	Location
Drug1	T1	L1
Drug2	T1	L1
Drug1	T3	L1
Drug1	T5	L1
Drug2	T6	L1
Drug2	T11	L2
Drug2	T12	L2
Drug3	T12	L3
Drug3	T15	L3

Table 8.1 is an example of raw RFID Data generated from 3 readers at different locations L1, L2, L3. As we can see from the table, the same patient, Patient1 has been detected three times in the same location L1 within a very short interval. Therefore, after we clean and merge all these duplicate Data into a single row, we get final output tuples in the form of (EPC, Time In, Time Out, Location), which is shown in Table 8.2.

Table 8.2 Cleansed RFID Data Sets for both Patients and Drugs tracking

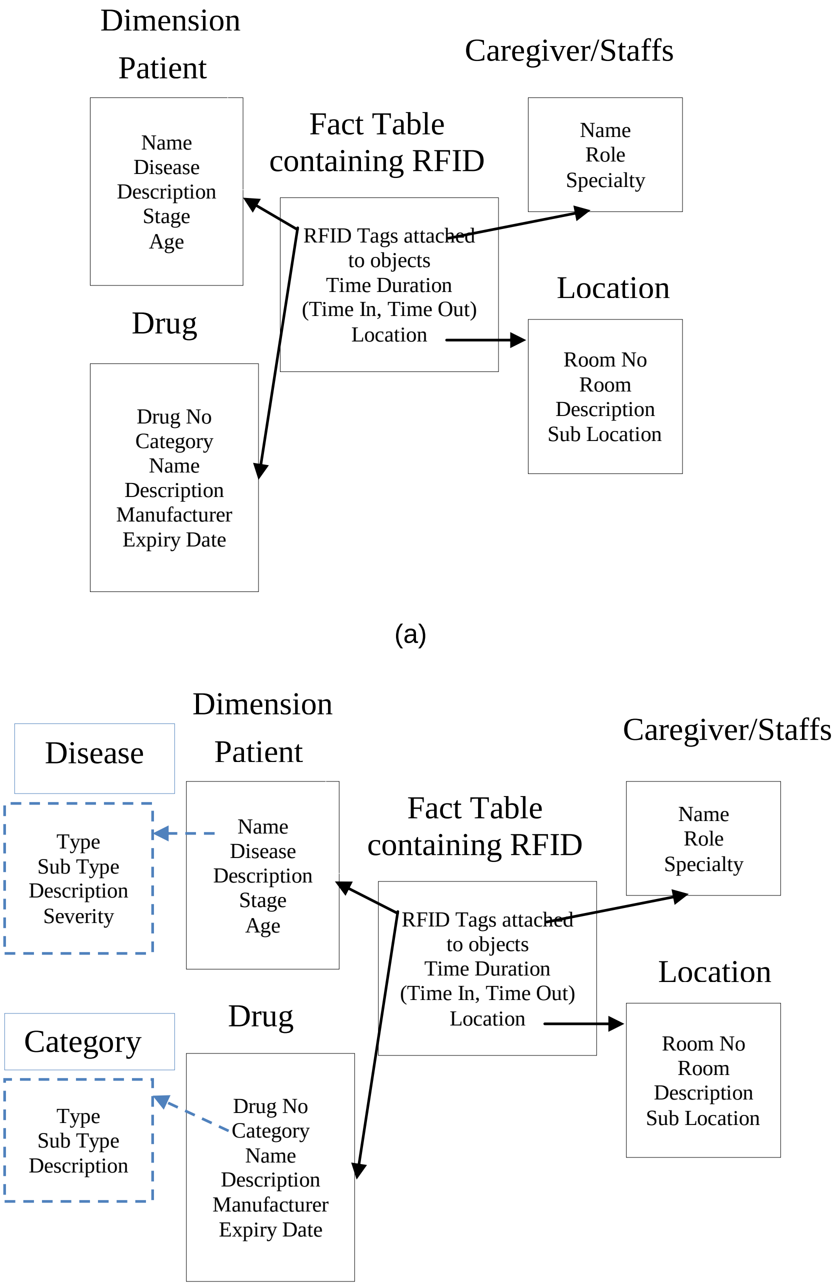
RFID Tag	Time In	Time Out	Location
Patient1	T1	T11	L1
Patient1	T15	T20	L3
Patient2	T1	T10	L1
Patient2	T11	T13	L2
Patient3	T12	T15	L2

Table 8.2 - *Continued*

RFID Tag	Time In	Time Out	Location
Drug1	T1	T5	L1
Drug2	T1	T6	L1
Drug2	T11	T12	L2
Drug3	T12	T15	L3

Here, the difference between Time Out and Time In, called Time Duration, represents the amount of time a particular object stays at that particular location without visiting any other intermediate location in between. The next step is to construct Data Cubes or Hyper Cubes by aggregating information across higher dimensions over these cleansed RFID Data. Any multidimensional storage model of Data Warehouse, such as Data Cube or Hyper Cube, consists of two types of tables. One type is the Dimension Tables that store path independent information for each type of RFID tagged objects or location. The other type of tables is called the Fact Table that stores observed or measured facts as well as contain pointers to the Dimension Tables. For example, the Dimension Table for Patient may contain different path-independent dimensions, such as Patient's Name, Age, Type of Disease, Stage, prescription and so on. Compared to Dimension Table, a Fact Table contains cleansed RFID Data, which are records of various time dependent observed events at various locations in our assistive environment.

However, the two common conceptual Data modeling schemes to describe these multi dimensional storage models are: 1) Star Schema that consists of a fact table with a single table for each dimension and 2) Snow Flake Schema, where dimension tables from Star Schema can be organized into hierarchy. Figure 8.2 displays examples of the corresponding Star and Snow Flake schemas related to our assistive environment. We construct Data Cubes or Hyper Cubes such as Patient-Drug Interactions by modeling and viewing Data in multiple dimensions of Patient (Name, Disease, Description, Prescription, Age), Drug (Drug No, Category, Name, Description, Manufacturer, Expiry Date), Time (Day, Week, Month, Quarter, Year), Location (Room No, Room, Description, Sub Location), a fact table containing pointers to the dimension tables and so on. Our top-level cuboids contain flat summary of all Patients, Drugs, Time and location Data, which are called as Apex Cuboids or 0-D Cuboids.



(a)

(b)

Figure 8.2 (a) Star Schema and (b) Snow Flake Schema

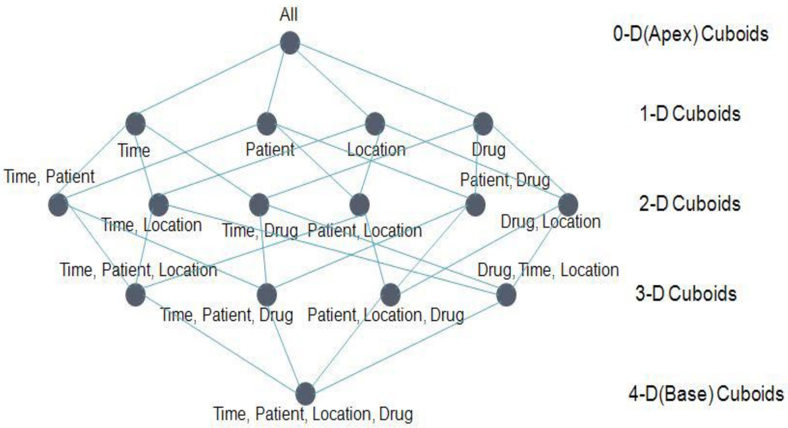


Figure 8.3 Example RFID Hyper Cube by aggregating Data up to 4 Dimensions

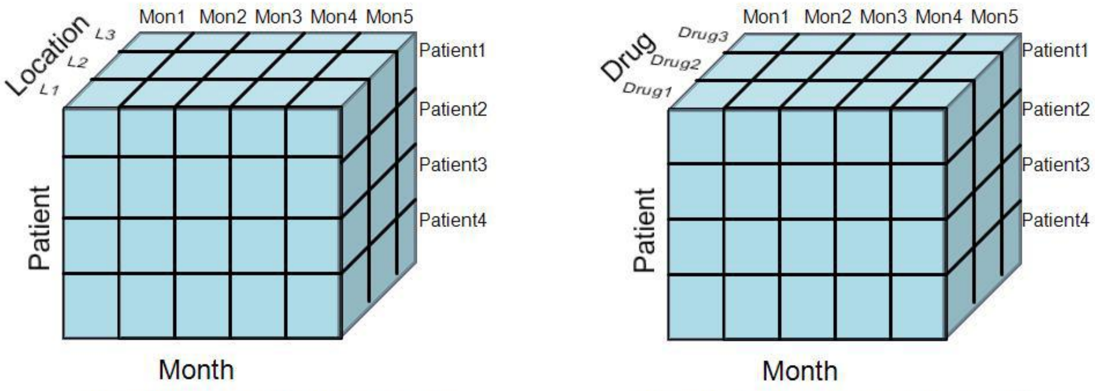


Figure: Location of Patient over time

Figure: Patient-Drug Interaction over time

Figure 8.4 Sample RFID Cuboids

As long as we aggregate these Data on higher level, we can construct higher dimensional Cubes. Figure 8.3 describes an example RFID Hyper Cube by aggregating Patient-Drug interaction Data up to 4 dimensions. Now, two example RFID Cubes for tracking location of Patients over Time (month) and possible Patient-Drug interactions over Time (month) are depicted in Figure 8.4. Finally, we can describe the RFID warehouse as a multi level database, where the lower level contains the raw RFID Data generated from the application, the next level contains cleansed RFID records, next contains minimum abstraction level RFID Cuboids and the top level consists of frequently queried RFID Cuboids or hyper cubes as in Figure 8.5.

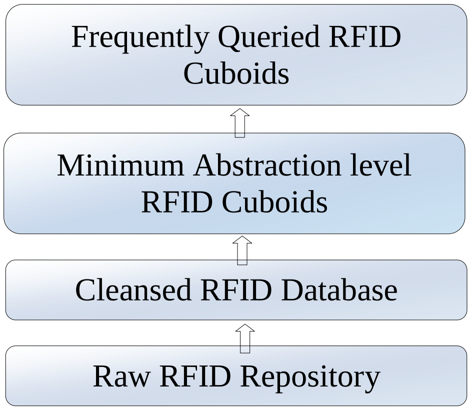

Frequently Queried RFID Cuboids
Minimum Abstraction level RFID Cuboids
Cleansed RFID Database
Raw RFID Repository

Figure 8.5 RFID Warehouse as a multi level Database

8.2 OLAP Operations

OLAP (Online Analytical Processing) is the most commonly used technique to analyze complex Data from Data Warehouses. Any multidimensional model of Data Warehouse can be displayed in hierarchical views. The most common operations in OLAP include Slice and Dice, Roll-up, Drill-down and pivot [83], which can be described as follows:

Slice and Dice: Displays subset of multi dimensional array by performing projections along dimensions.

Roll-up: Grouping or Summing up Data from the most detailed level to the most generalized level. In short, this operation aggregates Data into more coarse-grained view by computing larger units along a dimension. Examples of Roll-up display are: listing Patient's activities according to their disease type or summarizing daily, monthly or yearly Drug-intakes of different types of drugs according to their category. By aggregating the low level Data values, Roll-up eventually conceals these raw Data from public access and thus, can preserve Data privacy as well.

Drill-down: Opposite of Roll-up that displays Data into more detailed/specific level.

Pivot: Performs rotation. In a word, changes dimensional hierarchy or orientation of a Data Cube.

8.3 Sliding Window Techniques

Although Data Warehousing supports historical Data Analysis over RFID Data Stream, sometimes it is necessary to answer a query not over the entire history of Data, rather over the most recent Data [84]. One example scenario can be: “When a Patient takes the wrong medicine, the system should generate an alarm or report to a responsible doctor/nurse about this incident immediately.” In this scenario, it is not possible to first load all the arriving Data into DBMS (Database Management System) and then trigger the alert at real time. Because, a traditional DBMS does not support rapid and continuous loading of Data Items, thus making it difficult to support executing continuous queries like above. In order to detect abnormal or emergency behavior from this type of continuous Data Stream, a sliding window of the past ten to fifteen minutes or max one to two hours of Data needs to be considered to process the RFID data at real time. A Sliding Window is a very well known approximation technique to answer continuous queries efficiently. Here, the answer to a continuous query is completely based on what Data has been seen so far and can be changed/updated as soon as more Data arrives. Examples of one of these types of queries are Aggregate queries. Determining a suitable Window Size to provide fast responses is the key issue in Sliding Window techniques. If the window size is very large, it may not be possible to process all Data at once by keeping it into memory adding more delay. In the context of applying RFID technology to assistive environments, keeping the window size even in the range of one/two hours can be difficult to deal with, as the amount of Data generated in this very short period can be massive. But in real world, actual window size specification completely depends on the particular application. If the system needs to trigger an action immediately after a patient takes the wrong medicine or an unauthorized person carries the wrong drug, keeping the window size as small as possible does the best. Processing queries like Daily medicine intake or usage per patient requires updating the results from previous windows with more recent Data.

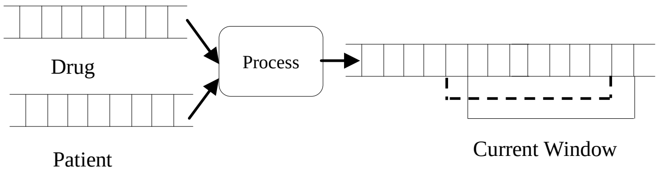


Figure 8.6 Applying Sliding Window over RFID Data

Figure 8.6 shows an example of processing a continuous stream of spatio-temporal Patient-Drug RFID Data using the Sliding Window mechanism. Each time, only the most recent events within the window of thirty minutes time interval is considered.

8.4 Performance Study

In this section, we do a performance analysis of the Data Warehousing technique over raw RFID Data generated from an assistive environment. For this experiment, we have generated millions of synthetic RFID Data for both Patients and Drugs movements by randomly constructing a set of different parameters (RFID Tag, Tag Detection Time and Location) for an individual RFID Event. Here, we have considered the movements and the interactions among ten patients and ten different drugs in five different locations at several times over seven days. This synthetic Data set conforms to the actual RFID Data set. Because in real environment, as soon as an object, either a person or a drug, carrying a RFID tag comes close to a reader, the reader detects that tag and transmit it's ID to the system. After receiving the ID, the system can record the current time of when the tag has been detected. Here, we are assuming that the actual time, when the tag has been detected by the reader is same as the time recorded by the system after its ID has been transmitted to the system. Moreover, the system can record the location information indirectly from the reader's location. From the synthetic RFID Data Set, we have constructed Cleansed RFID Database by eliminating redundancy and merging continuous RFID Events generated for an individual patient or a same drug in a single location into one single event. The merged event contains both 'Time in' and 'Time out' for the object in that particular location. Depending on the Randomness of the raw RFID Data Set, the size of the cleansed data set varies significantly. For a completely Random Dataset of 5000 raw RFID

Data, the size of the cleaned RFID Dataset found is 4019. For Datasets containing a little bias such as a dataset that contains continuous RFID events for an individual patient at one single location for a small time interval, which is the most likely real time scenario, the size of the cleansed Dataset was decreased to a great extent. Figure 8.7 shows the size of different cleansed RFID Data Sets against the size of corresponding Raw RFID Data Sets generated through our application.

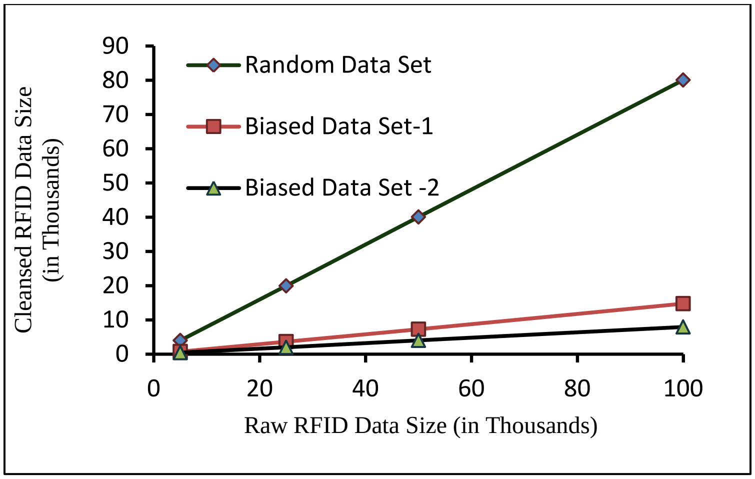


Figure 8.7 Cleansed RFID Data Set vs. Raw RFID Data Set

In the next step, we have computed different levels of RFID Cuboids from the Cleansed Data Set generated from the Random Data Set. Level-1 Cuboids contain the presence of Patients in different locations or activities of patients over time. Level-2 Cuboids consist of tracking activities of all Patients by aggregating information across days and locations. We have constructed Level-3 Cuboids by aggregating interactions of different Drugs and Patients along the three dimensions of locations, family of Drugs and Patients grouped by their disease categories. We have also computed Level-3 Cuboids for activities of Patients in different locations in different times by aggregating these information across the three dimensions of days, locations and Patients grouped by their disease categories. After we have computed the Cuboids and Hyper Cubes, we have executed several queries over both the cleansed RFID

Data Set and the RFID Cuboids. We have executed these queries on SQL Server 2009 installed on an Intel Core 2 Duo machine with 4GB RAM. Figure 8.8 shows the execution times for four different queries using Level-3 RFID Cuboids vs. Cleansed RFID Data Set. In each case, the Level-3 Cuboids take less time than the relational DBMS (Cleansed Data Set) by a noticeable amount.

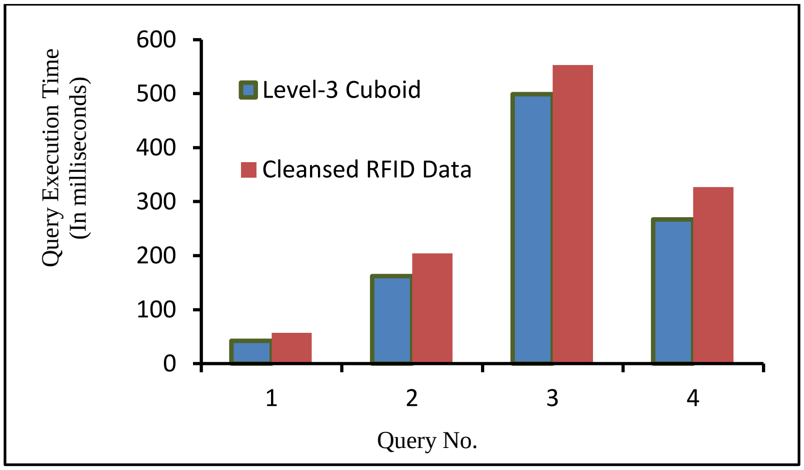


Figure 8.8 Query Execution Time (Level-3 Cuboids vs. Cleansed RFID Data)

Moreover, with the increase in the size of Data Set, the percentage improvement over the query execution times between RFID Cubes and Cleansed RFID Set increases more as depicted in Figure 8.9. In this run, we have executed a simple query of “listing all the activities of a single patient (P0) into one location (L0) from January 1 to January 4”. Since our fundamental assumption in querying into assistive environment is to deal with massive amount RFID Data generated through monitoring and tracking of several objects, significant performance enhancement can be achieved in this case by building Data Warehouse rather than querying over relational DBMS.

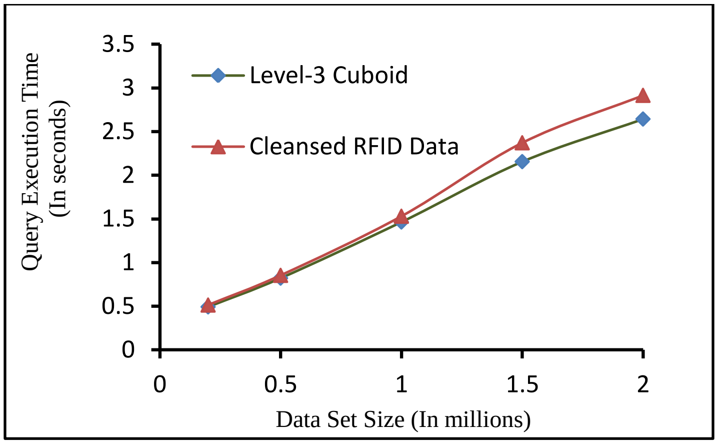


Figure 8.9 Performance improvement in query execution time with increasing Data Size

8.5 Discussions

Our system applies two well-recognized techniques: 1) Data Warehousing and 2) Sliding Window Protocol for answering both statistical and real-time queries using RFID in an Assistive environment. Our experimental result shows, the RFID Cuboids can boost performance to a great extent over the relational DBMS by considering the fact that Data generated using RFID applications in assistive environment is very large in volume. For the performance analysis, we have constructed our Data Warehouse from massive amount of synthetic Data generated through our application. However, we have only focused on Data Warehousing and OLAP-based query analysis of RFID Data. Other efficient methods for Data Mining over RFID Data such as trend analysis, path clustering, and outlier detection remains as another line of future work.

CHAPTER 9

FRAMEWORK TO INTEGRATE AND QUERY ARCHIVE DATA

9.1 Introduction

Data generated from pervasive applications can be of many different types, such as sensor readings, text, audio, video, medical records, etc. A pervasive application deploys various non-invasive sensors as an integral part of a persons' assistive daily living to automatically monitor his activities. It may also deploy a less-intrusive audio and video recording system, based on the needs and privacy requirements of the patient. To be effective, an assistive environment needs to store the data generated from pervasive applications into a common repository and to provide the healthcare providers a flexible and easier access to that repository.

However, current advances in sensor technology allow many different sensors that use different technology to be used interchangeably to generate and record similar information about an environment. For example, a person may wear a wireless wrist-watch [86] as a heart rate monitor, which may also include a 3-axis accelerometer, a pressure and a temperature monitor. On the other hand, a Sunspot sensor [87] can be used as an accelerometer and a temperature sensor as well. Although, both devices can be used to generate the same acceleration and temperature data, the format of their representation can be very different. In fact, a sensor can be programmed in many different ways to deliver data in different formats. For example, a sunspot can be programmed to transmit either 3-axis accelerometer data or the angle of acceleration. Therefore, even for a simple sensor device, such as a sunspot, the data storage may contain data in various schemas and configurations. As a result, it is very hard for a caregiver or a doctor to understand and remember each such different schema to query the data repository.

A pervasive application may sometimes refine a sensor reading specific to its settings. For example, sensor devices, such as a smoke/heat detector, only detect and transmit results when the environment reaches a predefined threshold. On the other hand, a stand-alone temperature sensor, such as a sunspot, can be configured to indirectly generate a heat alarm when the temperature exceeds some pre-specified heat threshold. Moreover, such thresholds and conditions can change dynamically depending on a user's query requirements, thus making it highly infeasible for the system to cache all possible answers related to a user's query beforehand.

As a solution to this problem, our system constructs a Digital Library that consists of a repository of sensor data collected from various pervasive applications and a flexible query interface for the user that requires minimum knowledge about the real metadata schemas from a user's point of view.

9.1.1 Framework

The framework for the sensor data repository contains datasets, such as C_{11}, C_{12}, …, C_{nm}, derived from many different sensors, collected over a long period of time, possibly after years of experiments [88]. We call each such dataset $C_{ij,}$ collected from various applications and contexts, a collection. Each collection is also associated with a metadata schema, which describes the format of that collection. For example, in Figure 1, the schema S1 describes the collection C11. A user can query over these different collections by simply using one of his preferred schemas, which we call a *virtual schema*, S_v in our framework. In our framework, given a user-query Q_v over the collections of different schemas, the user first provides the needed parts of "Map(S_i -> S_v)", the mapping for each individual schema S_i of a selected collection to the specified virtual schema, S_v. Given that the user specifies all such required mappings, the system next applies these mappings to return the final query answers, denoted by Q_1, Q_2…Q_n. Thus, based on our framework, a user may query on any kind and of any number of collections and obtain fine-grained results without knowing details about the real schemas. In the future, we are planning to use this framework as the building block to obtain the background knowledge for automatic metadata mappings.

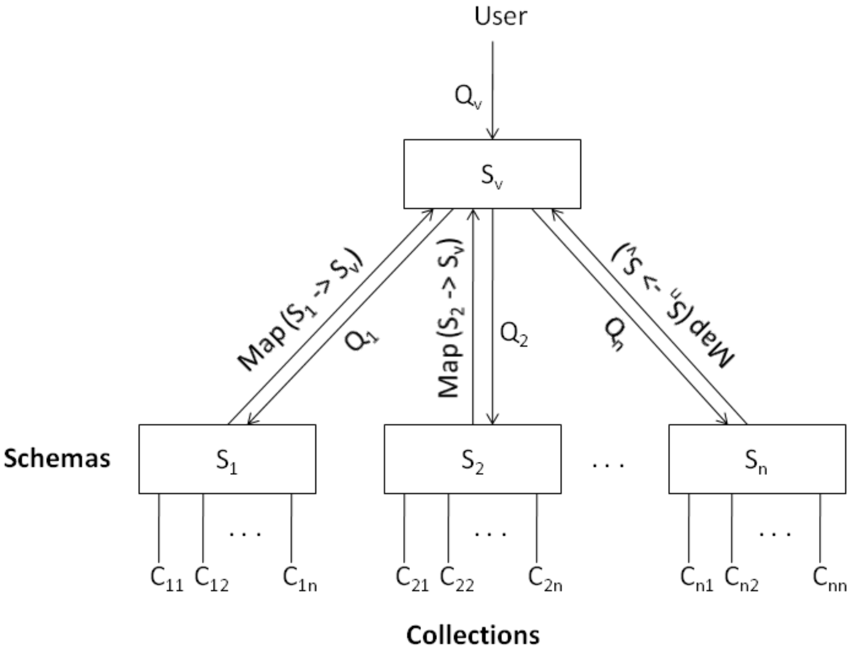


Figure 9.1 Framework to query over the Digital Library

9.1.2 Challenges

Sensor records may not of the same format, since they can be generated from different types of sensors installed in various setups, or different versions of similar sensors, with small variations on the record formats. Besides, the values recorded from similar sensors can have different semantics too. The following example shows that even for a simple scenario, such as monitoring "whether a door of an assistive apartment is open or not", the system may require many different mappings to answer a user query.

A pervasive application can use variety of door sensors, which eventually produces door-"open/close" datasets in different formats. For example, the system may use a sensor based on door-mounted magnetic contacts, which denotes that the door is open (1) or closed (0) as an OPEN_STATUS (1/0) attribute, combined with the time of this event as an attribute named TIME_DETECTED. Any accelerometer sensor, such as a sunspot, can also be mounted on the door to obtain similar information. Since, a sunspot can be preprogrammed in different configurations, one configuration may transmit and store Cartesian (x, y, z) coordinates of the door, while the other may compute and transmit the angle of the current position of the door. Such a programmable sunspot may provide both TIME_RECEIVED and TIME_BROADCAST attributes as well.

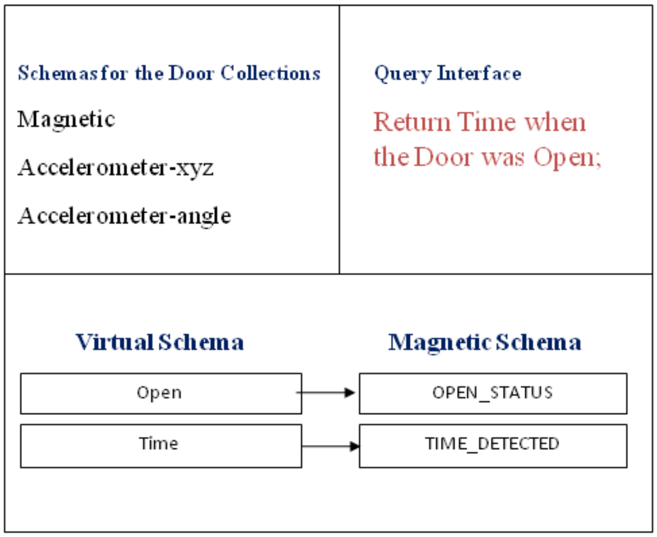


Figure 9.2 Metadata Mapping between similar attributes with different names

A user may not know anything about the schema for the sensor being used to collect such door-data, but he still may ask a query such as “Give me the time when the door was open” over the collection of door datasets. Although answering such a simple query seems trivial, the mappings can be very different and complex. Figure 9.2 describes the simplest mapping scenarios, where differently named attributes convey similar information, such as Open and OPEN_STATUS or Time and TIME_RECEIVED from the “Virtual” and the “Magnetic” schemas respectively.

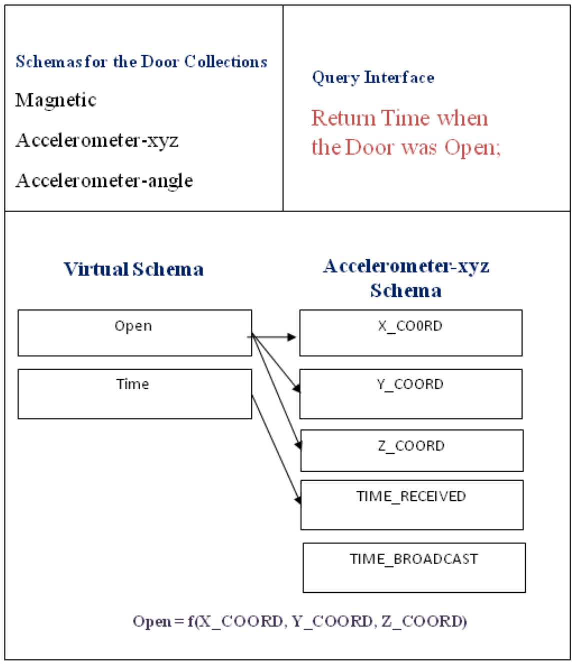


Figure 9.3 Metadata Mapping from One-to-Many

However, a metadata field from one schema can be mapped to multiple metadata fields from the other by applying appropriate mapping function. Hence, if a user wants to select such an one-to-many mapping, he needs to specify the correct mapping function as well to obtain meaningful results. Figure 9.3 shows one example scenario, where the field “Open” in the virtual schema is mapped as a function of X_COORD, Y_COORD and Z_COORD fields of the Accelerometer-xyz schema. Figure 9.4 shows another scenario, where a user may need to specify a condition even for a one-to-one metadata mapping. As shown in the example, a user may map the field “Open” to the field “Angle”. But, since the attribute “Angle” does not directly specify the open status of the door, the user also needs to specify a condition, such as “the door is open, if the angle is greater than at least 4 degrees” and so on.

However, even if the metadata field describing an attribute is the same for two schemas, the associated units of their values can be completely different. For example, the temperature can be described in either Celsius or Fahrenheit, distances can be written either in feet or in meters, time can be a represented in milliseconds or as a general date/time expression. Therefore, in addition to field mappings, the system needs to be able to apply value mappings as well.

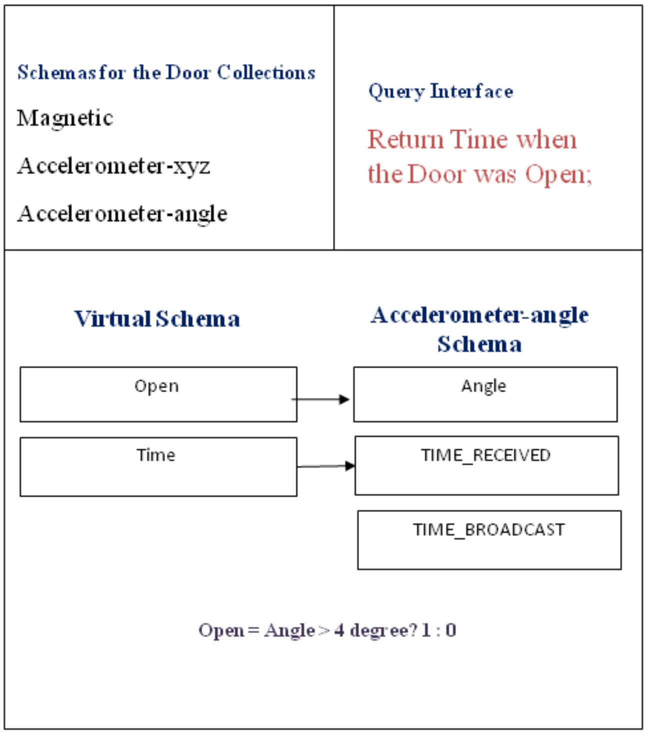

Schemas for the Door Collections
Magnetic
Accelerometer-xyz
Accelerometer-angle
Query Interface
Return Time when the Door was Open;
Virtual Schema
Accelerometer-angle Schema
Open
Time
Angle
TIME_RECEIVED
TIME_BROADCAST
Open = Angle > 4 degree? 1 : 0

Figure 9.4 Indirect Metadata Mapping

9.1.3 Contribution

The system provides a flexible query interface for searching relevant records from a Digital Library of sensor data. Such an interface requires minimum background knowledge and returns results in a format chosen by the user himself. The Library also stores the history of mappings as part of a user profile. Thus, a user can re-use existing mappings to query over the same collections multiple times, thus providing minimal information. However, the system is flexible enough to allow the user to re-write some existing mappings or to add new conditions to the previously specified mappings. Our system may also suggest existing mappings from other user's profiles, which helps a new user to get some idea about the mapping between schemas.

The flexibility of reusing and revising metadata mappings make the framework adaptive to different data management needs and different sensor metadata and formats.

9.2 Description of the System

9.2.1 Design

Based on the design, a user may have one of two different roles: he can either contribute or retrieve data to/from the repository. Whenever a user adds a new collection to the repository, he must also specify the metadata schema to describe that collection. Our system stores all such metadata schemas in the repository and stores the link between each collection and its corresponding metadata schema.

Our system also provides a suitable query interface for the users who want to search over the repository for relevant data. The query interface consists of a suitable window to browse for different collections and metadata schemas. The user can either select a metadata schema from the list of collections or may use a “virtual” schema, which is not used in any collection. The user may either query over the entire library by selecting all collections or select some specific collections. As soon as the user asks to execute a query, the system first checks the stored metadata mappings to see whether some mappings already exist in the system from the schemas of the selected collections to the preferred virtual schema. Our interface collects and displays all such mappings and asks the user to select any of the following mapping choices:

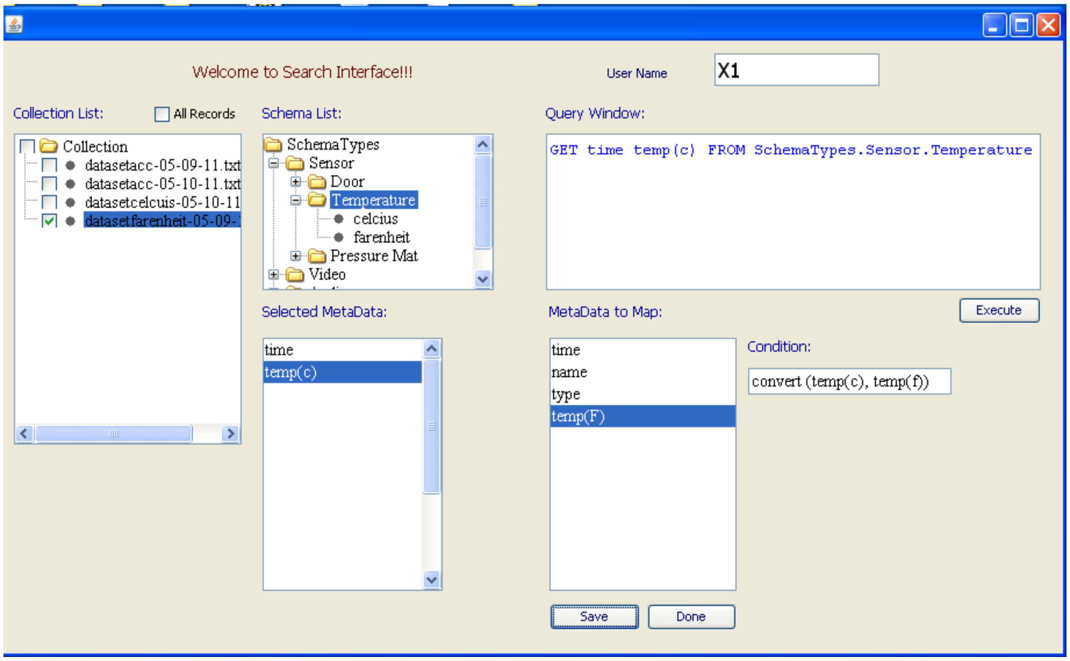
1. Re-use an existing mapping.
2. Select the mapping used last time, which is the default choice for the preexisting mappings.
3. Specify a new mapping. This is the default option for the mappings that have not been specified yet. Since, initially, the system does not have any stored mappings; our interface asks the user to enter the mappings first. A user may only provide mappings for the fields that he wants to query at that moment. However, he may enter a new mapping based on

similar attribute names or RDF descriptions. He may also define a function to map one schema attribute to the others.

4. Select the recommended mapping from the system. Since our system stores mapping preferences into a user profile, it can identify the most commonly used mappings by different users for a pair of schemas and can recommend such mappings to the user.

Next, as soon as the user specifies his preferred mappings, the system retrieves the resulting records or data from the collections and returns those to the user. The user may either view all such records in a separate output window or browse individual collection manually and only view the results for that particular collection.

9.2.2 Implementation



User Name: X1

□ All Records

Collection List:
- □ Collection
  - □ datasetacc-05-09-11.txt
  - □ datasetacc-05-10-11.txt
  - □ datasetcelcuis-05-10-11
  - ☑ datasetfarenheit-05-09-

Schema List:
- SchemaTypes
  - Sensor
    - Door
    - Temperature
      - celcius
      - farenheit
    - Pressure Mat
  - Video

```
GET time temp(c) FROM SchemaTypes.Sensor.Temperature
```

Selected MetaData: time, temp(c)

MetaData to Map: time, name, type, temp(F)

Condition: convert (temp(c), temp(f))

Figure 9.5 A Screenshot of the Query Interface

Our system incorporates a prototype interface in Java to store and view mappings and collections data from the Library. A screenshot of that interface is shown in Figure 9.5. From the interface, the user may express a query using the attributes listed under a particular metadata schema. Executing such a query is straightforward, as it does not require any mapping.

However, the user does not need to use any such schema to search over the collections. Instead, the user can select any type and number of collections and can query using a virtual schema. In this case, he may be asked to provide the right pair-wise mapping for each selected dataset of new metadata schema to his preferred virtual schema, before the query can be executed. The user may also specify a mapping condition using the interface and as soon as he saves it, the mappings become part of the user profile. However, whenever he is done with all the mappings, the system executes the query and returns the results to the user.

9.2.3 Management of Mappings

In our framework, a mapping, M_{ij}, from schema S_i to the schema S_j is associated with a set of bindings A_{jk}:$f_{jk}(A_{i1},...A_{in})$ that derive the attribute value of A_{jk} in S_j from the attributes $A_{i1},...A_{in}$ in S_i . The expression f_{jk} may consist of simple arithmetic operations, such as the comparison of an attribute value with a constant threshold, and string manipulation operations, such as string concatenation. These expressions are represented as abstract syntax trees and are stored in a mapping repository along with the rest of the mapping information. When a query is expressed in the schema S_i and the mapping M_{ij} is selected to query the data collections that conform to the schema S_j, then the query attributes are mapped to the S_j attributes using the expression f_{jk} and the derived query is used for querying the data collections that match S_j.

9.3 Discussions

In short, our system incorporates a framework for the repository of heterogeneous sensor data. The system also provides a prototype for a flexible query interface, which allows the user to search over the collections of different metadata schemas using minimal background knowledge. The system then collects and stores possible mappings from various users incrementally, which could work as a building block to derive commonly accepted mappings. As a future work, we are planning to use our mapping repository as a knowledge base to facilitate automatic metadata mappings.

CHAPTER 10

CONCLUSION

In this dissertation, we have presented a PeopleTrack system that tracks multiple people in a scene simultaneously. The system provides different views to remotely monitor the persons living in an assistive environment as well as to identify their activities and behavior. The system also collects and stores the spatio-temporal data generated from the deployed sensor networks to monitor and track the person living in the assistive environment. In addition, the system builds efficient data models on top of the Data Store, which makes it faster to respond to the historical queries. The Query Interface provided by the system offers a generic view of the collection of data stored in the Database, irrespective of their schemas.

Our result shows that combination of different passive sensors, which are less-intrusive, can be deployed in an assistive environment to identify and track activities. We have shown the effectiveness of such passive sensor technology to solve some important problems, such as medicine intake and depression assessment, which can be further generalized to other similar human-centered activities and conditions. Furthermore, we have shown that integration of data from heterogeneous sources, i.e. data from multiple modalities, improves the accuracy of the overall results.

REFERENCES

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

[2] P.R. Sun, B.H. Wang, and F. Wu, “A new method to guard inpatient medication safety by the implementation of RFID,” *Journal of Medical Systems,* vol. 32, 2008, pp. 327–332.

[3] R. Wimmer, M. Kranz, S. Boring, and A. Schmidt, “CapTable and CapShelf-Unobtrusive Activity Recognition Using Networked Capacitive Sensors,” *Networked Sensing Systems, 2007. INSS'07. Fourth International Conference on,* 2007, pp. 85–88.

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

[5] “Study on the requirements and options for RFID application in healthcare: Identifying areas for Radio Frequency Identification deployment in health care delivery: A review of relevant literature,” 2009.

[6] T. Teixeira, D. Jung, and A. Savvides, “Tasking networked cctv cameras and mobile phones to identify and localize multiple people,” in Proc. of 12th Ubicomp. ACM, 2010, pp. 213–222.

[7] R. Cucchiara, M. Fornaciari, A. Prati, and P. Santinelli, “Mutual calibration of camera motes and rfids for people localization and identification,” in *Proceedings of the ACM/IEEE ICDSC 2010,* Aug. 2010.

[8] Y. Chang, R. Yan, D. Chen, and J. Yang, “People identification with limited labels in privacy-protected video,” in *International Conference and Multimedia Expo*, 2006.

[9] T. J. Hazen et al., "Multi-Modal Face and Speaker Identification for Mobile Devices Multi-Modal Face and Speaker Identification for Mobile Devices ", in R. I. Hammoud, B. Abidi, and M.

Abidi, eds, *Face Biometrics for Personal Identification: Multi-Sensory Multi-Modal Systems*, Springer-Verlag, Heidelberg, Germany, April 2007.

[10] H. K. Ekenel, M. Fischer, Q. Jin, and R. Stiefelhagen, “Multi-modal person identi_cation in a smart environment,” CVPR Biometrics Workshop, Minneapolis, USA, June 2007.

[11-new] S. Ferdous, K. Vyas, and F. Makedon, "A survey on Multi person Identification and Localization", In proceedings of PETRA 2012, to appear.

[12] C. -H. Kuo and R. Nevatia, “How does Person Identity Recognition Help Multi-Person Tracking?,” In *CVPR 2011.*

[13] M. Andriluka, S. Roth, B. Schiele, “People Tracking-by-Detection and People Detection by Tracking,” In: CVPR’08. (2008).

[14] B. Leibe, K. Schindler, L. V. Gool, “Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles,” PAMI 30 (2008) 1683–1698.

[15] A. Ess, B. Leibe, K. Schindler, L. VanGool, “RobustMulti-PersonTrackingfromaMobile Platform,” PAMI 31 (2009) 1831–1846.

[16] B. Wu, R. Nevatia, “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Part Detectors,” IJCV 75 (2007) 247–266.

[17] D. Gavrila, S. Munder, “Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle,” IJCV 73 (2007) 41–59.

[18] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, “Multi-person tracking with sparse detection and continuous segmentation,” In *ECCV,* 2010.

[19] J. Krumm, S. Harris, B. Meyers, B. Brumitt, and M. H. and Steve Shafer, “Multi-camera multi-person tracking for easy living,” In *Third IEEE International Workshop on Visual Surveillance, July 1,* 1997.

[20] D. Focken, R. Stiefelhagen, *“Towards Vision-Based 3-D People Tracking in a Smart Room,”* IEEE International Conference on Multimodal Interfaces, Pittsburgh, PA, USA, October 14-16, 2002, pp. 400-405.

[21] D. Hahnel, W. Burgard, D. Fox, K. Fishikin, and M. Philipose, “Mapping and Localization with RFID Technology,” Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp.1015-1020, 2004.

[22] M. Bouet, & d. Santos, A.L., “RFID tags: Position principles and localization techniques,” Wireless Day 2008 (WD08), pp. 1-5, 2008.

[23] P. Wilson, D. Prashanth, and H. Aghajan, “Utlizing RFID signaling scheme for localization of stationary objects and speed estimation of mobile objects,” In *IEEE International Conference on RFID*, 2007.

[24] J. Brusey. M. Harrison, Ch. Floerkemeier. and M. Fletcher, “Reasoning about uncertainty in location identification with RFID,” In *IJCAI-2003 Workshop on Reoaoning with Uncenoirq in Robotics.* 2W3.

[25] J. Hightower, C. Vakili, G. Borriello, and R. Want, “Design and Calibration of the SpotON Ad-Hoc Location Sensing System,” Seattle, WA, August 2001.

[26] L. Ni, Y. Liu , Y. Cho Lau, A. Patil, “LANDMARC: Indoor Location Sensing Using Active RFID,” Wireless Networks 10, p.701–710, Kluwer Academic Publishers. Netherlands, 2004.

[27] D. Joho, C. Plagemann, W. Burgard, 2009, “Modeling RFID Signal Strength and Tag Detection for Localization and Mapping,” Proceedings of the IEEE International Conference on Robotics and Automation (ICRA): 3160-3165.

[28] T. Rappaport, “*Wireless Communications: Principles and Practice,”* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[29] M. M. Trivedi, K. S. Huang, and I. Mikic, “Dynamic context capture and distributed video arrays for intelligent spaces,” In IEEE Transactions on Systems: Man and Cybernetics - Part A: Systems and Humans, volume 35, January 2005.

[30] N. Fox, R. Gross, P. Chazal, J. Cohn, and R. Reilly, “Person Identification Using Automatic Integration of Speech, Lip and Face Experts,” In *ACM SIGMM 2003 Multimedia Biometrics Methods and Applications Workshop,* Berkeley, CA, 2003.

[31] T. Hazen, E. Weinstein, B. Heisele, A. Park, J. Ming, “Multi-Modal Face and Speaker Identification for Mobile Devices,” In: Hammoud, R.I., Abidi, B.R., Abidi, M. A. (eds.) Face Biometrics for Personal Identification: Multi-Sensory Multi-Modal Systems, pp. 123–138. Springer, Heidelberg (2006).

[32] L. Klingbeil and T. Wark, “A wireless sensor network for real-time indoor localization and motion monitoring,” in IPSN '08: Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008), Washington, DC, USA, 2008, pp. 39-50, IEEE Computer Society.

[33] J. A. Stankovic, et al, “Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges,” in High Confidence Medical Device Software and Systems (HCMDSS) Workshop, Philadelphia, PA, June 2-3, 2005.

[34] Dr. P. C. Jain, K. P. Vijaygopalan, “RFID and Wireless Sensor Networks,” Proceedings of ASCNT – 2010, CDAC, Noida, India, pp. 1 – 11.

[35] L. Zhang and Z. Wang, “Integration of RFID into wireless sensor networks: Architectures, Opportunities,” In *Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops (GCCW’06)*, Changsha, China, pp. 463–469*,* October 2006.

[36] A. Salah, R. Morros, J. Luque, C. Segura, J. Hernando, O. Ambekar, B. Schouten and E. Pauwels, (2008), “Multimodal identification and localization of users in a smart environment,” *JMUI,* 2(2): 75–91.

[37] H. Gonzalez, Jiawei Han, Xiaolei Li, and D. Klabjan, “Warehousing and Analyzing Massive RFID Data Sets,” *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on,* 2006, p. 83.

[38] K. Cotterill, “Distributed Patient-Drug Tracking in Healthcare Organizations,” *Solution White-Paper,* Bonsai Development Corporation, 2004-2005.

[39] F. Wang and P. Liu, “Temporal management of RFID data,” *Proceedings of the 31st international conference on Very large databases,* Trondheim, Norway: VLDB Endowment,

2005, pp. 1128-1139.

[40] D. Mahajan, N. Kwatra, S. Jain, P. Kalra and S. Banerjee, A Framework for Activity Recognition and Detection of Unusual Activities, ICVGIP, 2004, pp. 15-21.

[41] T. Teixeira, D. Jung, G. Dublon and A. Savvides, Recognizing Activities from Context and Arm Pose using Finite State Machines, ICDSC, August - September, 2009.

[42] L. Pelc and B. Kwolek, Activity Recognition Using Probabilistic Timed Automata, Pattern Recognition, I-Tech, 2008, pp. 119-135.

[43] A. Papangelis, G. Galatas, S. Ferdous, S. Kapidakis and F. Makedon, "Computer Aided Sensor Placement and Automatic State Definition", *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'10),* Article No. 65, *June 23-25, 2010.*

[44] S. Ferdous, L. Fegaras, and F. Makedon, "Applying data warehousing technique in pervasive assistive environment". PETRA 2010.

[45] A. Cangialosi, J. Monaly, and S. Yang, "Leveraging RFID in hospitals: Patient life cycle and mobility perspectives," *Communications Magazine, IEEE*, vol. 45, 2007, pp. 18-23.

[46] R. Weinstein, "RFID: a technical overview and its application to the enterprise," *IT Professional*, vol. 7, 2005, pp. 27-33.

[47] J. Woods, "RFID Enables Sensory Network Strategies to Transform Industries,", 2005.

[48] H. Gonzalez, Jiawei Han, Xiaolei Li, and D. Klabjan, "Warehousing and Analyzing Massive RFID Data Sets," *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, 2006, p. 83.

[49] R. Derakhshan, M. Orlowska, and Xue Li, "RFID Data Management: Challenges and Opportunities," *RFID, 2007. IEEE International Conference on*, 2007, pp. 175-182.

[50] F. Wang and P. Liu, "Temporal management of RFID data," *Proceedings of the 31st international conference on Very large data bases*, Trondheim, Norway: VLDB Endowment, 2005, pp. 1128-1139.

[51] J. K. Vinjumur, E. Becker, S. Ferdous, G. Galatas and F. Makedon, "Web based medicine intake tracking application", PETRA 2010.

[52] E. Becker, V. Metsis, R. Arora, J. Vinjumur, Y. Xu, and F. Makedon, “SmartDrawer: RFID-based smart medicine drawer for assistive environments,” *Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments*, 2009, p. 49.

[53] “Lab Precision and Analytical Weighing - METTLER TOLEDO - United States.”

[54] “Phidgets Inc. - Unique and Easy to Use USB Interfaces.”

[55] “jahmm - Project Hosting on Google Code.”

[56] S. Ferdous, G. Galatas, K. Vyas, E. Becker, L. Fegaras and F. Makedon, "Multi Person Identification and Localization in Pervasive Assistive Environments", IJCSMR journal, vol.1, issue 4, 2012, pp. 751-758.

[57] S. Ferdous, E. Becker, L. Fegaras and F. Makedon, "Multi-person identification and localization using RFID and passive sensor technology". PETRA 2011.

[58] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), 16(9), Sept. 2004.

[59] http://www.alientechnology.com/readers/alr9900.php.

[60] M. Wang, “A Flexible, Low-Overhead Ubiquitous System for Medication Monitoring.”

[61] D.L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, 1997, pp. 6–23.

[62] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In KDD-94: AAAI Workshop on Knowledge Discovery in Databases, pages 359-370, Seattle, Washington, July1994.

[63] “SunSPOTWorld - Home.”

[64] M. J. Herrmann, A. C. Ehlis, and A. J. Fallgatter, “Bilaterally reduced frontal activation during a verbal fluency task in depressed patients as measured by near-infrared spectroscopy”, Neuropsychiatry and Clinical Neuroscience 16(2):170-175, 2004.

[65] G. Salvadore, A. C. Nugent, H. Lemaitre, D. A. Luckenbaugh, R. Tinsley, D. M. Cannon, A. Neumeister, C. A. Jr. Zarate, and W. C. Drevets, “Prefrontal cortical abnormalities in currently depressed versus currently remitted patients with major depressive disorder”, Neuroimage,54(4):2643-2651, 2010.

[66] S. Ferdous, V. Metsis, G. Galatas, H. Huang, M. Basco, L. Fegaras and F. Makedon, "Automatic Depression Assessment from Speech Records", ICICTH 2011, Samos, Greece, July 14 - 16, 2011.

[67] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan“,Analysis of emotion recognition using facial expressions speech and multimodal information” In Proceedings of Int. Conf. Multimodal Interfaces, pp. 205–211, 2004.

[68] H. J. Go, K. C. Kwak, D. J. Lee, and M. G. Chun, “Emotion recognition from facial image and speech signal”, In Proceedings of Int. Conf. Soc. Instrument and Control Engineers, pp. 2890–2895, 2003.

[69] S. Hoch, F. Althoff, G. McGlaun, and G. Rigoll, “Bimodal fusion of emotional data in an automotive environment”, In Proceedings of ICASSP, vol. 2, pp. 1085–1088, 2005.

[70] Z. Zeng, J. Tu, M. Liu, and T. S. Huang, “Multi-stream confidence analysis for audio–visual affect recognition”, In Proceedings of Int. Conf. on Affective Computing and Intelligent Interaction, pp. 946–971, 2005.

[71] X. Zhe and A.C. Boucouvalas, “Text-to-emotion engine for real time internet communication”, In Proceedings of Int. Symposium on Communication Systems, Networks, and DSPs, Staffordshire University, pp. 164–168, 2002.

[72] T. S. Polzin and A. Waibel, “Emotion-sensitive humancomputer interfaces”, In Proceedings of ISCA Workshop on Speech and Emotion, pp. 201–206, 2000.

[73] B. Schuller, R. Muller, M. Lang, and G. Rigoll, “Speaker Independent Emotion Recognition by Early Fusion of Acoustic and Linguistic Features within Ensembles”, In Proceedings of Eurospeech - Interspeech, pp. 805–809 2005.

[74] F. Metze, A. Batliner, F. Eyben, T. Polzehl, B. Schuller and S. Steidl, “Emotion Recognition using Imperfect Speech Recognition”, In Proceedings of Interspeech, pp. 478-481, 2010.

[75] I. Nouretdinov, S. G. Costafreda, A. Gammerman, A. Chervonenkis, V. Vovk, V. Vapnik and C. H. Y. Fu, “Machine learning classification with confidence: Application of transductive conformal predictors to MRI-based diagnostic and prognostic markers in depression”, NeuroImage, 56(2):809-813, Multivariate Decoding and Brain Reading, 2011.

[76] S. G. Costafreda, C. H. Y. Fu , M. Picchioni , T. Toulopoulou , C. McDonald , E. Kravariti , M. Walshe , D. Prata , R. M. Murray and P. K. McGuire, “Pattern of neural responses to verbal fluency shows diagnostic specificity for schizophrenia and bipolar disorder”, BMC Psychiatry, pages 11-18, 2011.

[77] J. F. Cohn, T. S. Kruez, I. Matthews, Y. Yang, M. H. Nguyen, M. T. Padilla, F. Zhou and F. D. L. Torre, “Detecting Depression from Facial Actions and Vocal Prosody “, In Proceedings of Affective Computing and Intelligent Interaction, pp. 1-7, 2009.

[78] H. Liu and R. Setiono. “Chi2: Feature selection and discretization of numeric attributes”, In Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, pp. 88, USA, 1995.

[79] F. Nie, H. Huang, X. Cai, and C. Ding, “Efficient and robust feature selection via joint 2, 1-norms minimization”,In Proceedings of NIPS, 2010.

[80] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines”, Machine Learning 46(1/3): 389-422, 2002.

[81] C. Cortes and V. Vapnik, “Support-vector network”, Machine Learning, 20:273-297, 1995.

[82] N. Cristianini and J. Shawe-Taylor, “An introduction to support Vector Machines: and other kernel-based learning methods”, Cambridge University Press, 2000.

[83] R.A. Elmasri and S.B. Navathe, *Fundamentals of Database Systems*, Addison-Wesley Longman Publishing Co., Inc., 1999.

[84] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, Madison, Wisconsin: ACM, 2002, pp. 1-16.

[85] S. Ferdous, Z. Le, L. Fegaras, and F. Makedon, "Tagging and Warehousing for Health Data", *Proceeding of the 8th International Conference on Information Communication Technologies in Health* (ICICTH'10), Samos, Greece, July 15-17, 2010.

[86] Temperature, Wireless, USB Data Logger or Watch Development Tool, http://focus.ti.com/docs/toolsw/folders/print/ez430chronos.html?DCMP=Chronos&HQS=Other+OT+chronos#technicaldocuments.

[87] Sun SPOT World, http://www.sunspotworld.com/.

[88] S. Ferdous, S. Kapidakis, L. Fegaras, and F. Makedon, "An Adaptable Framework for Integrating and Querying Sensor Data", EANN/AIAI(2), 2011, pp. 430-438.

[89] C. McMurrough, S. Ferdous, A. Papangelis, A. Boisselle, and F. Makedon, "A Survey of Assistive Computing Devices for Cerebral Palsy Patients", In proceedings of PETRA 2012, to appear.

[90] S. Ferdous, E. Becker, F. Makedon, "Human Activity Monitoring Instrument for Use as a Bioinstrumentation", Proceedings of the 27TH Southern Biomedical Engineering Conference (SBEC), April 29-May 1, 2011, Arlington, TX.

BIOGRAPHICAL INFORMATION

Shahina Ferdous received her B.Sc. in Computer Science and Engineering from the Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology (BUET) in 2006. She worked as a Software Engineer in Therap BD Ltd. during 2007-2008, until she joined the BS to PhD program at University of Texas at Arlington (UTA) in 2008. She joined the Heracleia Human-Centered Computing Laboratory at UTA as a Research Assistant in 2009. Her research interests include Data Mining, Database, Sensor Network, Pervasive Assistive Applications. Ms. Ferdous has co-authored several peer reviewed papers published in the technical conferences and journals. She has also served as a committee member and reviewer in many conferences.